RF Project 762945/714250
Semi-Annual Report

AN EXPERIMENTAL STUDY OF AN ULTRA-MOBILE
VEHICLE FOR OFF-ROAD TRANSPORTATION

Robert B. McGhee and Kenneth J. Waldron
College of Engineering

20000 803080

For the Period
April 1, 1983 - September 30, 1983

DEFENSE SUPPLY SERVICE
Washington, D.C. 20310

Contract No. MDA903-82-K-0058

S DTIC
ELECTE
MAY 2 1 1984
E

May, 1984

**The Ohio State University
Research Foundation**
1314 Kinnear Road
Columbus, Ohio 43212

84   05 21   019

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-$A14132$-3	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* AN EXPERIMENTAL STUDY OF AN ULTRA-MOBILE VEHICLE FOR OFF-ROAD TRANSPORTATION		5. TYPE OF REPORT & PERIOD COVERED Semi-Annual Technical Report 4/1/83 - 9/30/83
		6. PERFORMING ORG. REPORT NUMBER 762945/714250
7. AUTHOR(s) Robert B. McGhee and Kenneth J. Waldron		8. CONTRACT OR GRANT NUMBER(s) MDA903-82-K-0058
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Ohio State University Research Foundation, 1314 Kinnear Road Columbus, Ohio 43212		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DEPARTMENT OF THE ARMY, Defense Supply Service Room ID-245, The Pentagon Washington, D.C. 20310		12. REPORT DATE May, 1984
		13. NUMBER OF PAGES 301
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Walking machine
Robotics
Off-road vehicles
Legged locomotion

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report summarizes research accomplished during the fourth six-month period of this three-year project. The research is concerned with the design and construction of an experimental vehicle making use of actively-controlled suspension units. The current vehicle concept envisages locomotion without the use of wheels or tracks, but rather employs six active suspension elements, each providing support and propulsion analogously to a biological leg. The report describes a prototype hydraulically-powered vehicle leg making use of an

(continued)

DD $_{1\ JAN\ 73}^{FORM}$ 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Block 20 (Abstract) - Continued

innovative hydrostatic system with one variable displacement pump for each actuator. This system is designed to permit locomotion at a considerably lower energy cost than would be possible with a conventional valve-controlled configuration using pressure-compensated pumps. The report also discusses hydraulic safety features based on explosively-actuated emergency valving to either lock actuators or bypass pumps.

Electrical system research includes validation of control software for four vehicle operational modes. These modes range from a semi-automatic mode in which the operator controls either the body or individual legs, to a fully automatic terrain-following mode in which the operator merely steers the vehicle body with a three-axis controller while the on-board computer controls both body attitude and foot placement. Experimental results presented include the use of proximity sensors for automatic control of ground clearance during swing phases of leg motion.

Semi-Annual Technical Report
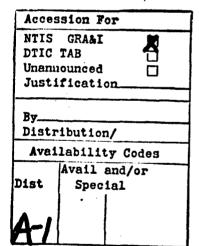
for

DARPA Contract MDA903-82-K-U058

AN EXPERIMENTAL STUDY OF AN ULTRA-MOBILE
VEHICLE FOR OFF-ROAD TRANSPORTATION

Prepared by

Robert B. McGhee and Kenneth J. Waldron

covering the period

April 1, 1983, through September 30, 1983

College of Engineering
The Ohio State University
Columbus, Ohio 43210

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	

By	
Distribution/	

Availability Codes	
Dist	Avail and/or Special
A-1	

1. INTRODUCTION

Research under this contract began as scheduled on October 1, 1981. During the first twenty-four months, no significant deviations occurred from the plan contained in our proposal dated May 13, 1981. Current project milestones are summarized in a list dated October 28, 1983, and attached to our Quarterly R & D Status Report for the period October 1, 1983 through December 31, 1983. All milestones on this list pertaining to the period covered by the present report have been met on schedule. The following pages present some details concerning work relating to each of these milestones. Further information is provided in the attached appendices and in theses, dissertations, published papers, and videotapes previously furnished to DARPA. A list of these items is presented in Appendix 1. Numerical references in the following text refer to this list.

2. MECHANICAL SYSTEM RESEARCH

2.1 Hydraulic System Design for Precision Footing

During this period, a completely new hydraulic actuation system using a hydrostatic principle was designed and major components were selected. In support of this effort, the breadboard leg actuation system was reconfigured as a hydrostatic system and tested for dynamic response and other performance characteristics. A major motivation for this effort was the unsuitability of the earlier tested valved configuration for operation in a precision footing mode. Additional motivation was provided by the high projected energy consumption in the drive and lift actuator circuits resulting from large variations in load and velocity conditions seen by those actuators during each leg cycle.

Testing of the breadboard leg with the reconfigured power transmission and actuation system indicated that satisfactory dynamic performance should be

1

achieved by a properly optimized hydrostatic actuation system. These tests were terminated after determination of the dynamic response characteristics of the pump controllers and the leg actuation systems to make way for installation of a prototype of an optimized system as described below.

The configuration of the optimized hydrostatic actuation system is presented in Appendix 2. Major design decisions taken during this period included selection of the pumps to be used, and of the rotary actuators which control the pump displacements. Several competing control valve types will be tested on the prototype leg to determine which is most suitable. Specially configured actuators were designed in detail during this period and were manufactured by a local supplier. The design includes many novel features: a light-weight aluminum barrel with hard faced anodizing, oil feed through the rods, including feed of both ends of the drive actuators from one end of the rod, etc. A schematic of one of the drive actuators is included in Appendix 2. A complex design task which was also completed in this period was that of a drive train to power pumps mounted both on the leg and on the body of the vehicle. Appendix 3 includes a schematic drawing of this system.

2.2 Preliminary Foot Design

It has become apparent, during consideration of foot action, that it is highly desirable to have the foot parallel to the ground immediately before and during touchdown, and to maintain the foot flat on the ground during the complete contact period in most locomotion situations. Therefore, a system was designed to maintain the sole of the foot approximately parallel to the body at all times. A schematic drawing of this system is included in Appendix 3. It is a passive hydraulic system in which a master cylinder acts between the drive-actuator roller assembly and the intermediate link of the leg

2

pantograph. The master cylinder is connected via a closed hydraulic system to a slave cylinder which acts between the leg shank and the foot. Accumulators are included in the circuit to allow rotation with controlled compliance about the ankle joint. There will also be stops to prevent excessive ankle rotation. This system maintains the foot parallel to the vehicle body to within 3° throughout the working volume of the leg.

The foot was designed as a support for a sole unit which may include a three axis force sensor. Actuator pressure sensing may be used as an alternate means of inferring foot forces. Experience with the breadboard leg and the requirements of the foot attitude maintenance system led to the adoption of a single axis hinge at the ankle in place of the spherical joint and spring combination used in the earlier foot design.

2.3 Prototype Leg

A prototype vehicle leg was designed and constructed by the end of this period. Some structural components from the breadboard leg were incorporated into this unit while others were completely redesigned. A major part of the system is the new hydrostatic drive train, power transmission, and actuation system described in Section 2.1. Likewise, the new foot design and foot attitude maintenance system described in Section 2.2 is incorporated. A completely new shank has been built. This includes a cranked lower segment which simulation studies have shown to be necessary when crossing large obstacles. It is also designed to accommodate the new foot design and the passive foot attitude control system. The upper leg box has been substantially modified by the addition of mounting brackets for the new actuators and for the drive and lift pumps of the hydrostatic system. In fact, of the major structural components, only the two thigh links have been retained in their original form.

3

The prototype leg, shown in Fig. 1, is highly modular in comparison to the earlier leg design described in our previous Semi-Annual Report. The entire actuation system is mounted either to the leg structure itself, or to immediately adjacent portions of the test stand. This modularity will greatly simplify construction and maintenance of the vehicle.

2.4 Procedures for Maintaining Stability Under Normal and Emergency Conditions

Work under this heading included two major efforts. One of these was a theoretical and simulation study of maintenance of static stability by a six-legged vehicle when crossing obstacles and operating on slopes. The major results of this study are included in Appendix 3 and in the M.S. thesis of Shih-Liang Wang [20]. This work has strongly contributed to the understanding of these situations necessary for the design of vehicle coordination software and for the selection of operator strategies in difficult conditions.

The second effort is the design of hardware to permit the vehicle to respond rapidly to emergency situations. As a result of a joint OSU/Battelle study of the hardware requirements for this purpose, Battelle is working on a design for an explosively actuated valve system which will permit either locking or bypass of appropriate actuators in emergency situations.

2.5 Design of a Modular Cockpit Structure

It became apparent, early in this reporting period, that it would be difficult to accommodate all system hardware within the confines of the vehicle frame. At the same time, the division of responsibility for the cockpit structure (OSU) and for controls and instrumentation (U. Wisconsin) created coordination problems. Further, there was concern that the operator's vision of the feet and of the terrain on which they were to be placed was

4

Figure 1: Prototype leg mounted on test stand.

insufficient. It was, therefore, decided to abandon the concept of accommodating the cockpit within the main vehicle frame and instead to design a cockpit module which would be bolted to the front end of the frame. A design for the structure of such a module was completed during the reporting period. It will be manufactured early in the next reporting period. It was not possible to fully accommodate the specifications of the University of Wisconsin with regard to cockpit space because of the substantial weight penalty incurred by enlarging the cockpit frame. After extensive discussion, a compromise was reached on the dimensions of the cockpit module. Consideration will be given to retro-fitting a larger cockpit if the dimensions of this design prove to be inadequate during vehicle testing. Figure 2 is a photograph of the 1/10 scale ASV-84 mechanical configuration model showing the modular cockpit mounted to the main vehicle frame.

3. ELECTRICAL SYSTEM RESEARCH

3.1 Vehicle Operating Software

In our Semi-Annual Report dated February, 1983, preliminary definitions for a total of six ASV-84 operational modes were provided. Since that time, these modes have been further developed and related to the partitioning of the vehicle control software so as to permit efficient implementation on a multi-processor control computer. Details of this work are provided in Appendix 4. However, generally speaking, it can be said that the definitions of the six operational modes have not been changed in any significant way. Rather, the major accomplishment of the past six months has been the actual realization of four of the six modes on either the PDP-11/70 computer or the breadboard ASV-84 computer. These experiments are described briefly in the following paragraphs.

6

Figure 2: ACV-84 mechanical configuration model on terrain board.

The Utility mode has been simulated using the Intel Series III Microprocessor Development System (MDS). A VT-100 terminal has been interfaced to this system and used as an operator's console. A smaller CRT system suitable for mounting in the vehicle has been identified along with the necessary incerface cards. With this arrangement, the Coordination Computer described in Appendix 4 can be used analogously to an MDS system for control and field programming of the vehicle on-board computer.

A Precision Footing control mode has been realized on the PDP-11/70 computer. Physical testing with the OSU Hexapod has demonstrated its effectiveness. Details are provided in the following Section 3.2.

Omnidirectional control in the Close Maneuvering mode has been demonstrated on the PDP-11/70 using a kinematic model of the ASV-84 displayed on a vector graphics terminal. This simulation study includes the feature of automatic regulation of body pitch and roll attitude to accommodate terrain slopes in such a way as to maximize vehicle stability and maneuverability. A videotape showing real-time operation of this system has been provided to DARPA.

The Battelle and OSU portions of the software for the Terrain Following mode of control have been successfully integrated and tested on the breadboard ASV-84 computer. In this test, the PDP-11/70 computer was used to simulate the terrain, the ERIM scanner, and the vehicle. The resulting motion of the vehicle relative to the terrain was presented in real time on a vector graphics terminal. A shaded color graphics representation of this motion will be prepared using single-frame motion picture photography during the next six months of research.

At present, we do not plan to develop cruise and dash mode software under this contract, since both of these modes involve high speed motion which can only be undertaken after thorough field testing of the above four low speed modes. Instead, we intend to develop a plan for further vehicle testing and programming in the form of a proposal to DARPA for a continuation of the research of the current contract. This proposal will be completed and submitted for evaluation within the next six months of this program.

3.2 Demonstration of Precision Footing Control

It is recognized that, in circumstances involving very large geometrical obstacles or extremely rough terrain, fully automatic modes of body and leg coordination for the ASV-84 may prove to be inadequate. To deal with such situations, per the discussion in the preceding section of this report, a semi-automatic precision footing mode of control will be provided. In this mode, the operator will be able to take direct control of the body or of any of the six legs of the vehicle in Cartesian coordinates. A specific form of this control mode has been developed and evaluated using the OSU Hexapod as a test bed. Figure 3 shows one such test in progress.

The general characteristics of the currently operational precision footing control mode are as follows:

1. All control is accomplished with a single three-axis joystick.

2. The operator uses a keyboard to dynamically assign the joystick to one of eight functions: control of any one of the six vehicle legs, body attitude control, or body position control.

3. All joystick deflections are interpreted as rate commands. Specifically for any leg, fore-aft deflection controls fore-aft foot velocity, lateral deflection controls lateral foot velocity, and twist controls vertical
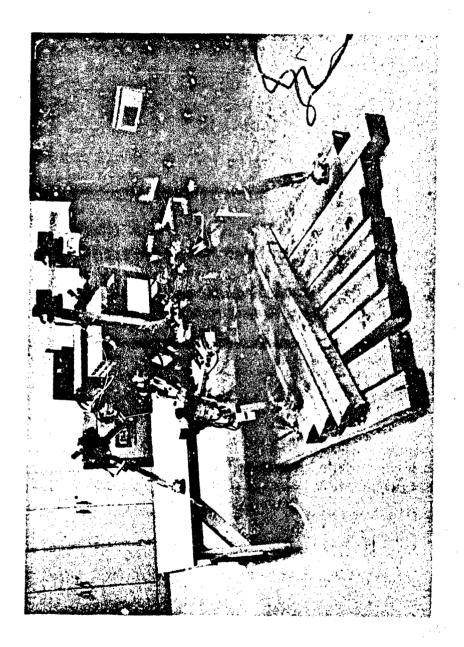
9

Figure 3: OSU Hexapod vehicle showing ability to overcome large obstacles in precision footing control mode.

foot velocity. These velocities are all interpreted in the body coordinate system. Body position control functions identically. For body attitude control, fore-aft deflection controls pitch rate, lateral deflection controls roll rate, and twist controls yaw rate.

4. When any one of the eight joystick assignments is invoked, the control computer coordinates joint motions so that the desired rates are achieved and all other rates are zero relative to Earth coordinates. Thus, during body motion, the feet of supporting legs as well as those of legs in the air remain stationary with respect to the terrain. This characteristic of precision footing control allows the operator to concentrate his attention on movement of the body with respect to the terrain, without concern for potential collisions of legs with obstacles.

As reported in [5], we have previously developed and tested safety software to prevent both body instability and collisions between adjacent limbs in any control mode. For the ASV-84, use of this safety software in conjunction with precision footing control will result in the vehicle refusing to execute maneuvers which could be harmful to it. In addition to this feature, a display of stability margin and of feet which can be safely lifted will assist the operator in avoiding the generation of unacceptable commands. The development of such displays will be undertaken during the next six month period of this project.

Laboratory evaluation of the above described precision footing control mode has shown that it is effective and relatively easy to learn. As was previously discovered in our studies of insect locomotion [33], "in phase" wave gaits seem to provide the best stepping patterns for overcoming large obstacles Such gaits utilize support by at least five legs at all times. The cycle of limb motions involves first placing the front feet, then the middle

11

feet, and finally the back feet. A videotape showing this gait and documen-
ting the experiment illustrated by Fig. 3 has been furnished to DARPA previous
to this report.

With more advanced hand controllers, it may be possible and desirable
for an operator to control more than three vehicle degrees of freedom at any
given time. This issue will be studied after suitable hand controllers have
been delivered to us by the University of Wisconsin. It is anticipated that
at least one such controller will be provided during the next six-month period
of this research.

3.3   Control of Foot Clearance by Proximity Sensors

The OSU Hexapod Vehicle has been fitted with an ultrasonic proximity
sensor on each of its six legs. Using these sensors, the vehicle has walked
over laboratory terrain with automatic control of foot clearance using proxi-
mity data. Full details of this experiment are provided in the attached
Appendix 5. The following paragraphs present the principal results of this
study.

The first finding was that, as expected, laboratory terrain produces
more or less specular reflection of ultrasound pulses; that is, since the wave
length of the sources used (40 kHz) is long compared to irregularities in the
tile floor of the laboratory or in the wooden obstacles shown in Fig. 3,
insufficient energy is reflected back to the receiver unless the incidence
angle of the transmitted sound is close to vertical. Practically, with care-
ful receiver design, it was found that angles up to about ± 20 degrees from
vertical can be tolerated.

The second finding was that leg trajectories can be tailored so as to
maintain the angular orientation of the lower limb segments of the OSU Hexapod

12

well within the ± 20 degree cone of tolerance of the ultrasonic sensors during locomotion over laboratory terrain. Under these conditions, foot clearance is regulated very accurately without prior knowledge of terrain elevation. Example trajectories from such tests are presented in the attached Appendix 5. A videotape including a demonstration of foot altitude control by proximity sensing has been previously furnished to DARPA.

The third major result of these experiments is that single frequency transducers do not function reliably on natural terrain when the average size of soil particles or rocks is comparable to or greater than the wavelength of the ultrasound illumination. This result was obtained from soil bin tests and can also be anticipated qualitatively, since such terrain presents the possibility of destructive interference of reflected energy at the receiver. The experiments detailed in Appendix 5 demonstrate that this is such a serious problem that single frequency ultrasound proximity sensing is impractical for natural terrain. It now appears that a multi-frequency or "chirp" source is required. In addition, an array of such sensors may be needed on each leg to assure adequate ground coverage, since each sensor has a rather narrow beamwidth. Five Polaroid ranging systems have been purchased to permit experimental evaluation of such a multi-frequency array on the prototype ASV-84 leg. Initial testing of this system is planned for the next six month period of this research.

13

Appendix 1

List of Publications

RESEARCH PUBLICATIONS SUPPORTED BY CONTRACT MDA903-82-K-0058

College of Engineering
The Ohio State University
Columbus, Ohio 43210

The following theses, dissertations, and papers have been produce⁴ with

the support of DARPA Contract MDA903-82-K-0058 since its inception on October

1, 1981. Copies of most of these documents are available upon request.

1.   Waldron, K.J., Song, S.M., Vohnout, V.J. and Kinzel, G.L., "Computer-Aided Design of a Leg for an Energy-Efficient Walking Machine," Pro-ceedings of the 7th Applied Mechanisms Conference, Kansas City, December 7-9, 1981, pp. VIII-1 to VIII-8.

2.   Vohnout, V.J., Mechanical Design of an Energy Efficient Robotic Leg for Use on a Multi-Legged Walking Vehicle, M.S. thesis, The Ohio State University, June, 1982.

3.   Waldron, K.J., Frank, A.A., and Srinivasan, K., "The Use of Mechanical Energy Storage in an Unconventional, Rough-Terrain Vehicle," 17th Intersociety Energy Conversion Engineering Conference, Los Angeles, California, August 8-13, 1982.

4.   Brown, F.T., Dynamic Study of a Four-Bar Linkage Walking Machine Leg, M.S. thesis, The Ohio State University, August, 1982.

5.   Ju, J.T., Safety Checking System with Voice Response for the OSU Hexapod, M.S. thesis, The Ohio State University, August, 1982.

6.   Pugh, D.R., An Autopilot for a Terrain-Adaptaive Hexapod Vehicle, M.S. thesis, The Ohio State University, August, 1982.

7.   Tsai, C.K., Computer Control Design of an Energy-Efficient Leg, M.S. thesis, The Ohio State University, August, 1982.

8.   Srinivasan, K., Waldron, K.J., and Dworak, J.A., "The Design and Evaluation of a Hydraulic Actuation System for a Legged Rough-Terrain Vehicle," presented at ASME Winter Annual Meeting, Phoenix, November, 14-16, 1982, Robotics Research and Advanced Applications, ASME, 1982.

9.   Kao, M.L., A Reliable Multi-Microcomputer System for Real Time Control, M.S. thesis, The Ohio State University, December, 1982.

10. Chang, T.W., Motion Planning for Locomotion of a Multilegged Robot over Uneven Terrain, M.S. thesis, The Ohio State University, December, 1982.

11. Cheng, T.T., Computer Simulation of the Dynamics and Control of an Energy-Efficient Robot Leg, M.S. thesis, The Ohio State University, December, 1982.

12. Chung, T.S., Kinematic Simulation of an Adaptive Suspension Vehicle, M.S. thesis, The Ohio State University, December, 1982.

13. Barrientos, C.E., Development of a Multiple Microprocessor-Based Control System for Legged Locomotion Processing Using Interactive Design Tools, M.S. thesis, The Ohio State University, December, 1982.

14. Chuang, J.Y., Simulation of Load-Time History of an Adaptive Walking Vehicle by the Use of Rate Regulation, M.S. thesis, December, 1982.

15. Tsai, S.J., An Experimental Study of a Binocular Vision System for Rough Terrain Locomotion of a Hexapod Walking Robot, Ph.D. dissertation, May, 1983.

16. Gardner, J.F., Modeling and Control of the Hydraulic Drive System for a Walking Machine Leg. M.S. thesis, The Ohio State University, June, 1983.

17. Broerman, K.R., Development of a Proximity Sensor System for Foot Altitude Control of a Terrain-Adaptive Hexapod Robot, M.S. thesis, The Ohio State University, August, 1983.

18. Klein, C.A., Olson, K.W., and Pugh, D.R., "Use of Force and Attitude Sensors for Locomotion of a Legged Vehicle over Irregular Terrain," International Journal of Robotics Research, Vol. 2, No. 2, Summer, 1983, pp. 3-17.

19. Ozguner, F., Tsai, S.J., and McGhee, R.B., "Rough Terrain Locomotion by a Hexapod Robot Using a Binocular Ranging System," Proc. of First International Symposium on Robotics Research, Bretton Woods, N.H., September, 1983.

20. Wang, S.L., The Study of a Hexapod Walking Vehicle's Maneuverability Over Level Ground and Obstacles and Its Computer Simulation, M.S. thesis, The Ohio State University, September, 1983.

21. Song, S.M., Waldron, K.J., and Kinzel, G.L., "Computer-Aided Geometric Design of Legs for a Walking Vehicle," Proceedings of 8th Applied Mechanisms Conference, St. Louis, Missouri, September 19-21, 1983, pp. 70-1 to 70-7.

22. Vohnout, V.J., Alexander, K.S., and Kinzel, G.L., "The Structural Design of the Legs for a Walking Vehicle," Proceedings of 8th Applied Mechanisms Conference, St. Louis, September, 1983, pp. 50-1 to 50-8.

23. Gardner, J.F., Dworak, J.A., Srinivasan, K. and Waldron, K.J., "Design and Testing of a Digitally Controlled Hydraulic Actuation System for a Walking Vehicle Leg Mechanism," Proceedings of 8th Applied Mechanisms Conference, St. Louis, September, 1983, pp. 2-1 to 2-7.

18

24. Orin, D.E., Tsai, C.K., and Cheng, F.T., "Dynamic Computer Control of a Robot Leg," Proceedings of IECON '83, San Francisco, California, November, 1983.

25. Alexander, K.E.S., The Structural Design and Optimization of a Walking Machine Frame and Leg, M.S. thesis, The Ohio State University, December, 1983.

26. Lee, W.J., A Computer Simulation Study of Omnidirectional Supervisory Control for Rough-Terrain Locomotion by a Multilegged Robot Vehicle, Ph.D. dissertation, March, 1984.

27. Ozguner, F. and Kao, M.L., "A Multimicroprocessor System for Fault Tolerant Control of an Articulated Mechanism," IEEE Trans. on Industrial Electronics, May, 1984.

28 McGhee, R.B., "Vehicular Legged Locomotion," Advances in Automation and Robotics, ed. by G.N. Saridis, Jai Press, Inc., Greenwich, Conn., 1984.

29. McGhee, R.B., Orin, D.E., Pugh, D.R. and Patterson, M.R., "A Hierarchically-Structured System for Computer Control of a Hexapod Walking Machine," Proc. of Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, June, 1984.

30. Waldron, K.J., Song, S.M., Wang, S.L. and Vohnout, V.J., "Mechanical and Geometric Design of the Adaptaive Suspension Vehicle," Proc. of Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, June, 1984.

31. Waldron, K.J., Vohnout, V.J., Pery, A. and McGhee, R.B., "Configuration Design of the Adaptaive Suspension Vehicle," International Journal of Robotics Research, Vol. 3, No. 2, Summer, 1984.

32. Ozguner, F., Tsai, S.J., and McGhee, R.B., "An Approach to the Use of Terrain Preview Information in Rough-Terrain Locomotion by a Hexapod Walking Machine," International Journal of Robotics Research, Vol. 3, No. 2, Summer, 1984.

33. Pearson, K.G., and Franklin, R., "Characteristics of Leg Movements and Patterns of Coordination in Locusts Walking on Rough Terrain," International Journal of Robotics Research, Vol. 3, No. 2, Summer, 1984.

34. Orin, D.E., Tsai, C.K. and Cheng, F.T., "Dynamic Computer Control of a Robot Leg," accepted for publication in IEEE Trans. on Industrial Electronics.

35. Ozguner, F. and Tsai, S.J., "Design and Implementation of Binocular Vision System for Locating Footholds of a Multilegged Walking Robot," accepted for publication in IEEE Trans. on Industrial Electronics.

19

Appendix 2

Configuration Design of the Adaptive Suspension Vehicle

CONFIGURATION DESIGN OF THE ADAPTIVE SUSPENSION VEHICLE

by

Kenneth J. Waldron,
Vincent J. Vohnout,
Arie Pery

and

Robert B. McGhee

Ohio State University
Columbus, Ohio 43210

Abstract

The selection of vehicle and leg configuration and of power transmission and actuation system configuration for the Adaptive Suspension Vehicle (ASV) are discussed. The Adaptive Suspension Vehicle will be a proof of concept prototype of a proposed class of transportation vehicles for use in terrain which is not passable to conventional vehicles. It uses a legged locomotion principle. The machine will not be an autonomous "robot" in the sense that it will carry an operator. It will, however, have a very high level of machine intelligence and of environmental sensing capability. Much of the technology involved is unique and has potential for application to future robot systems. In this paper major aspects of the vehicle and leg geometry, the on-board processing configuration, and the hydrostatic power transmission system, are discussed.

1.   Introduction

The Adaptive Suspension Vehicle (ASV), which is presently under construction, is a vehicle intended to provide mobility over very rough terrains. It uses a statically stable legged locomotion principle rather than a wheeled or tracked principle. Figure 1 shows a 1/10th scale model of the machine on a terrain board. The motivation for building such a machine is that about half the Earth's surface is inaccessible using wheeled or tracked vehicles [1]. As compared to legged locomotion, wheeled locomotion is only attractive on prepared or naturally hard and even surfaces. In principle, legged locomotion should be mechanically superior over a considerable range of soil conditions [2] and is certainly superior when crossing obstacles. This is well illustrated by the fact that the majority of the terrain cited as inaccessible to wheeled and tracked vehicles in reference [1] presents little difficulty to animals using legged locomotion.

In view of the above, it is natural to ask why the legged locomotion principle has almost never been used in man-made vehicles. There are two reasons:

Firstly, in order to take full advantage of the legged locomotion principle over uneven ground it is necessary that the machine have sufficient independently controllable degrees of freedom in each leg to allow the body motion to be completely decoupled from irregularities met in the terrain over which it travels. We call such a machine "fully terrain adaptive." This requires a minimum of three degrees of freedom in each leg and, hence, twelve independently controllable degrees of freedom in a four-legged machine and eighteen in a six-legged machine. Four legs is the minimum number necessary for static stability. Prior to the very recent advent of compact, powerful microcomputers

automatic coordination of the motions of such a large number of degrees of freedom by means of an on-board controller was impossible. Manual coordination was attempted using sophisticated teleoperator technology in the GE Quadruped project [3]. The conclusions to be drawn from that attempt was that this approach is of marginal efficacy. It was possible to control the machine effectively to perform sophisticated movements. On the other hand, the coordination task placed great demands on the operator. A long period of training was necessary and operators became fatigued after a relatively short working period.

Secondly, efficient and well controlled delivery of power to an oscillatory motion system such as a leg is very much more difficult than driving a continuously rotating wheel. This situation is exacerbated by the large number of degrees of freedom to be independently powered, in contrast to the single degree of freedom powered in most wheeled vehicles or the two powered degrees of freedom in most tracked vehicles. In the past, quite a number of experimental legged vehicles have been built. Most of these are discussed, or at least referenced, in reference [4] and will, therefore, not be discussed here. A few will be mentioned at appropriate points in this article. However, almost without exception, fully terrain-adaptive walking vehicles have had very poor mechanical efficiency. Thus, the promise of superior mechanical performance on unprepared surfaces has certainly not been realized in these machines. The reasons for this, and the measures being taken to overcome them in the design of the ASV, form a substantial portion of the present article.

The ASV will be 5 m long, will stand about 3.3 m high at normal walking height and will have a track of 1.6 m. It will weigh about 2600 kg. Cruise speed is to be 2.25 m/s with sprint capability to 3.6 m/s. The machine will be completely self-contained. It will carry an operator and a 225 kg payload.

Power will be supplied by a 900 cc. motorcycle engine driving an energy
storage flywheel of capacity C.25 kw hr. Line-shafts driven from the flywheel
shaft will run down both sides of the vehicle. The leg degrees of freedom
will be actuated by a fully hydrostatic power transmission system. The
actuators are hydraulic cylinders. Each actuator is coupled to a variable
displacement pump powered from the line-shaft by means of a toothed belt.
This system will also be described in greater detail later in the article.

Although the machine will not operate autonomously, it will
carry very advanced sensing, computer processing and coordination systems.
In most operating conditions the operator's role is strategic. That is, the
operator commands direction and velocity but does not directly control foot
placement or body attitude. The computational system will consist of thirteen
Intel 86/30 single-board computers. Six of these will be leg controllers, four
will process information from the terrain scanner and generate choices for foot
placement. One will control the cockpit displays and transmit data from the
controller to the coordination computer, one will be the coordination computer
and one will be a backup computer itoring system operation and otherwise
assisting the coordination computer. Each of the 86/30 boards carries an 8086
microprocessor, an 8087 floating point and trigonometric co-processor and a
varying quantity of RAM. Nine of the 8086 microprocessors will be coupled via
a multibus. However, only one will control the multibus communications. Further
details of this system will be reported in future publications.

The machine will also have sophisticated sensory systems. A vertical
gyroscope will provide a vertical reference and rate gyroscopes and accelerometers
will provide information on body movements. All six legs will be fitted with
three-axis force sensors to determine the three force components at the "ankle."

Proximity sensors will also be fitted to the feet. However, the most sophisticated sensing system will be the terrain scanner. This operates on a very different principle from conventional robot vision systems. It is, in fact, an optical raaar operating in a CW mode with sinusoidal amplitude modulation. A beam of infrared light generated by a gallium arsenide laser is swept across the terrain in a raster pattern. Phase comparison of the modulation envelope of the reflected and projected beams yields a direct range measurement to the point on the ground struck by the beam. The effective range is determined by the ambiguity interval and is 10 m. Resolution at maximum range is about 40 mm. At a speed of 2 m/s each point will be viewed approximately five times before passing under the vehicle. The terrain scanner has been built by ERIM [5]. Battelle Columbus Laboratories are responsible for the coiputer and software for processing the terrain scanner information and feeding 'oothold selection information to the coordination computer.

A hand controller with at least three continuously controlled degrees of freedom will be used by the operator to command direction and speed of motion. Forward rate, lateral rate and yaw rate will be continuously controlled. Control of walking height and body pitch and roll will be via settable position controls. A CRT display will be used to allow the operator to access data on a variety of systems in a variety of operational modes. The University of Wisconsin is responsible for design of the controls and cockpit displays.

2. Performance Evaluation

Locomotion systems are subject to large variations in operating conditions. When moving on level ground, a locomotion system is very lightly loaded, at least at low speed. In contrast, when moving up a gradient the system may have to produce a large power. Mechanical efficiency, the usual measure of power train performance, is not useful when discussing vehicles. When moving at constant speed on a level road a vehicle is producing no output work so it has zero efficiency.

A much more useful parameter when discussing vehicle performance is the specific resistance, ϵ, defined as:

$$\epsilon = \frac{P}{WV}$$

where P is the mechanical power input to the vehicle; that is, the output power of the prime mover, W is vehicle weight and V is vehicle velocity. Specific resistance can also be thought of as the inverse of lift-to-drag ratio where "drag" is an effective drag including all energy dissipation mechanisms.

Figure 2 is adapted from Gabrielli and Von Karman [6]. It is a plot of specific resistance versus speed for a variety of vehicles on a log-log scale. It may be seen that biological legged locomotion compares well with wheeled systems on prepared surfaces. It may also be seen that artificial legged systems such as the GE Quadruped [3] and OSU Hexapod [7] do very poorly. The reasons for this and the means used to overcome them will be discussed in Section 3. The ASV leg has a specific resistance of 0.08 in laboratory tests. The overall vehicle specific resistance will be considerably higher because

of losses higher in the power train and because of overhead power needed to operate the computer etc. After careful analysis of these losses and power requirements it is projected that the ASV will have a specific resistance of about 0.3 as indicated on the diagram. It may be seen that this is competitive with tracked vehicles.

In order to characterize obstacle crossing performance it has been decided to make use of a small set of standardized obstacle geometries. Obstacle crossing is primarily a problem of vehicle geometry but involves considerations of traction force and power also. Extensive simulation studies have been performed to optimize vehicle geometry and determine the maximal obstacles which it can cross [8]. The machine will be geometrically capable of operating on a 60% grade, crossing a 1.8 m vertical sided ditch, crossing a 1.7 m vertical up or down step and crossing a 1.4 m vertical isolated wall. It will be capable of squatting on belly skids for parking and as an emergency maneuver.

3. Leg Geometry

The choice of six legs is a compromise between stability, fa..ored by larger numbers of legs, and simplicity. The ASV is designed to operate in a statically stable mode or, more accurately, in a quasi-statically stable mode in which the stability criterion is modified to include the effects of dynamic load transfer. The implications of static stability are discussed in some detail in reference [4]. If the duty factor is defined as the fraction of the cycle time each foot is on the ground (assumed to be the same for all feet) stability increases with increasing duty factor. At a given duty factor stability increases with increasing number of legs. However, the incremental gains diminish with increasing leg number. Thus the gain in stability in going from six legs to eight is much smaller than that in going from four legs to six [4].

The major limit on speed in most legged locomotion systems, including biological ones, is the time required to return the leg through the air to its starting position. There is a minimum time for this "transfer phase" in any legged locomotion system. The minimum duty factor for a four legged statically stable system is 0.75 [4]. For a six legged system it is 0.5. It is 0.375 for an eight legged system. If the same leg characteristics are assumed; that is, if the same return time, τ, and leg stroke, S, are assumed, the duty factor, β, affects the velocity, V, as follows:

$$V = \frac{S}{\beta T}$$

where T is the cycle time: the time for one complete motion cycle of all legs. The time available for leg return is evidently:

$$\tau = (1-\beta)T$$

Hence if τ is fixed, T is related to β by

$$T = \frac{\tau}{1-\beta}$$

and

$$V = \frac{S}{\tau} \left(\frac{1-\beta}{\beta}\right) \quad .$$

Thus, for a four legged system:

$$V = 0.333 \frac{S}{\tau} \quad ,$$

for a six legged system:

$$V = \frac{S}{\tau}$$

for an eight legged system:

$$V = 1.67 \frac{S}{\tau} \quad .$$

Thus, for a statically stable system, there are substantial speed and stability advantages in increasing the number of legs. This must be weighed against the increasing mechanical and computational complexity which is also entailed by increasing the number of legs. Moreover the assumption of constant leg stroke S implies a linear increase in body length with increasing numbers of legs.

As was noted in Section 1, a minimum of three degrees of freedom are required in each leg. The objective of leg design is to provide the maximum working volume for the foot with minimum leg structure. Thus the leg is similar in its kinematic role to the regional structure of a manipulator. The leg can be viewed as a manipulator minus the wrist degrees of freedom. Since the regional structure of a manipulator is optimized by having the two most inboard joint axes intersect at right angles, the third joint axis parallel to the second and the second and third members of equal length [9], it follows that this arrangement is also optimum for a leg.

Given the above basic leg geometry, there are two advantageous ways of mounting the leg to the body. The first axis may be mounted vertically or it may be mounted horizontally parallel to the longitudinal axis of the body. The first is the configuration used in the OSU Hexapod [7], the Sutherland and Sproull hexapod [10], the Tokyo Institute of Technology quadruped [11], the Moscow State University hexapod [12] and, in modified form, in the Odetics hexapod [13]. It is the horizontal first axis configuration, however, which has been chosen for the Adaptive Suspension Vehicle and the GE Quadruped [3]. The first configuration is broadly similar to typical insect or reptilian legs. The latter is closer to a mammalian leg plan. The reasons for favoring these configurations are primarily based on efficient use of energy. Avoidance of energy wastage is vital in a self-contained vehicle which must carry its own energy supply.

The advantage, and it is an important one, of having a vertical azimuth axis for the first joint is that, during a walking stroke, the large displacement is borne by that vertical joint. The other joints have relatively small displacements. The significance of this is that little work is done against gravity. The large displacement takes place about an axis parallel to the weight force and hence, when walking on the level, involves no gravitational work. The motions about the other two joints do require positive or negative work done against gravity but, since the displacements are small, these components are also small.

The primary disadavantage of a vertical first axis is that, when performing an oscillatory motion, the kinetic energy of the leg must be absorbed at the end of each forward or backward stroke. In a vertical swinging leg, the leg can operate as a pendulum converting kinetic energy to gravitational potential

energy at the end of each stroke and then converting it back after changing direction. This is not possible in a leg with a vertical first axis. Thus, the kinetic energy is usually wasted by conversion to heat in systems with legs of the first kind. This can be a serious problem in electrically actuated systems such as the OSU Hexapod [7] because of the relatively large kinetic energies stored in rapidly rotating motor armatures. Use of springs to store the energy as strain energy is, of course, feasible. However, storage of energy as strain energy becomes less attractive as the size of the machine increases since kinetic and gravitational potential energy scale as the fourth power of lineal dimension while strain energy scales as only the third power.

The vertical swinging leg given by a horizontal first joint also has significantly lower bending moments in the upper leg structure. It allows a narrower track, and hence frontal area, which is an advantage when pushing through vegetation.

The second type of leg has, however, a major disadvantage. This can be understood with the help of Figure 3 which shows two positions of the leg relative to the machine when the machine is walking straight and level. Only the second and third joints are shown since the first joint is not active for this motion. When the leg is in the forward position the directions of rotation about both joint axes are clockwise. The moment of the foot reaction is clockwise about the "knee" but counter-clockwise about the "hip." Thus, the actuator of the hip joint is doing positive work against the load but the actuator of the knee joint is absorbing work done on the joint by the load. That is, it is acting as a brake.

In the second position shown in Figure 3, the rotation about the hip joint is still clockwise, but that about the knee joint has reversed and is now

counter-clockwise. The moment of the foot reaction is clockwise about both joints. Thus, the knee joint actuator is now doing work against the load while the hip actuator is acting as a brake.

The important thing to realize is that, when moving on a level surface a locomotion system is doing no work provided frictional resistance is neglected. Thus, in both of the situations above, one actuator is doing work and the other is absorbing the same work! If the work absorbed is converted into heat, as in most conventional actuation technologies, this represents a problem.

It was noted that biological systems with this leg geometry do not suffer from this problem. However, their actuation geometry is much more sophisticated involving actuators which operate across two or more joints in addition to those which operate across only one [14]. This suggested use of different actuation arrangements and led to several linkage leg designs. These have been thoroughly discussed in references [15] and [3]. However, they proved to be difficult to realize in mechanical hardware. An alternative mechanism which achieves the necessary objective of de-coupling vertical and horizontal displacements is shown schematically in Figure 4. It is a planar, two degree of freedom pantograph hinged about a horizontal joint parallel to the longitudinal axis of the body. It's action is very similar to that of the second leg type described above. It is not the first use of a pantograph in a walking machine leg but differs from that of Hirose [11] in being a hinged planar pantograph, rather than a spatial pantograph, and from that of Kessis [16] in being a right pantograph, rather than a skew pantograph. The geometric optimization of the pantograph leg mechanism with respect to workspace size and shape is described in reference [8]. Structural design of this leg mechanism is discussed in reference [17].

4.   Power Transmission and Actuation

The Adaptive Suspension Vehicle will weight 2600 kg. When in the tripod
gait [4] which will be used for cruise, each leg carries half the vehicle weight
at some stage in the gait cycle. Of course, the load may be somewhat greater
in some conditions. The legs are being designed for a normal working load of
1400 kg. At the same time, it is essential to minimize the weight of the power
transmission and actuation system. This leads to a demand for a very high
ratio of actuator force to actuator weight. Using current technology, this
demand can only be met by a hydraulic actuation system.

There are two basic types of hydraulic power transmission and actuation
system which might be used. The first is the familiar valve-controlled system
with regulated supply. The second is the hydrostatic system type in which each
actuator is directly coupled to a variable displacement pump. The hydrostatic
type of system is attractive because the control of the actuator is accomplished
by non-dissipative means. Also, hydrostatic systems are capable of regenerative
operation allowing power to be stored during braking. Nevertheless, it was felt
that the valve controlled system type would be superior in dynamic response and
lighter in weight. Therefore studies were initiated to develop suitable valve-
controlled hydraulic circuit configurations for the actuators [18]. It was early
decided to group similar actuators on all six legs into the same circuits. The
reason for this is that the supply pressure requirements of similarly placed
actuators are similar and losses are reduced if the supply pressure is held to a
minimal margin over the highest needed actuator pressure drop. An interesting system
which uses some similar principles, and some different, to attack the problems
discussed below is described in Reference [19] and used on the six-legged walking
machine described in Reference [10].

It soon became apparent that the design of the actuator circuits would
present substantial difficulties. In order to understand this problem it is
appropriate to examine the leg-lift and drive actuator operating conditions.

The leg lifting and lowering actuators are subject to two very different operating conditions during each cycle. When the foot is on the ground, the actuator is under high load but is displacing at a low rate; ideal conditions for a pressure regulated valve-controlled system. However, when the leg is being returned, the leg lift actuator is lightly loaded and is subject to large displacement rates. Since a high supply pressure must be maintained to service actuators which are under contact loads, the light loading, and hence actuator pressure, of the actuator in a returning leg requires a large pressure drop across the servo valve controlling it. Multiplied by the large flow through the valve required by the large displacement rate, this pressure drop produces a very large rate of conversion of hydraulic system energy to heat. In fact, our estimates indicated that, even for mild terrain roughness, 15 to 20 kw would be used by the leg lift circuit. Since, in straight line motion, regardless of gradient, the leg lift actuators do no net work against gravity, virtually all of this energy would be converted to heat. This would, in turn, create severe oil cooling problems. Many different strategies were evaluated for overcoming this problem. The only ones which appeared to be in any way viable were the use of a second circuit to supply the actuators at appropriate pressure during the return phase, or the use of a two-ratio mechanical change transmission between the actuator and the joint. Neither is attractive. The second circuit would require an additional pump, a second set of supply lines, a set of switching valves and, very likely, a second set of control lines. The switching operation would introduce transients and control problems. Development of a suitable dual ratio mechanical transmission was judged to be very difficult within the time restrictions of the project schedule.

The difficulties in design of the drive circuit were also a product of the varied operating conditions seen by the actuators at different times in the motion cycle and in different modes of operation. It is anticipated that the drive actuators will mostly be comparatively lightly loaded during operation. When walking on the level, the drive actuators work against only frictional loads. Nevertheless, provision must be made for large loads when walking up steep gradients and for braking when walking downhill. A conventional system with fixed supply pressure would be very inefficient because the supply pressure would have to be set to accommodate maximum load, leading to large valve pressure drops when operating on the level or on moderate grades. A simple strategy to overcome this would be to put the supply pressure under computer control so that it could be held to the pressure needed by the heaviest loaded actuator plus a margin for valving. Unfortunately, the valving margin must be substantial to give good control resolution. A second alternative is to regulate flow rather than pressure. This is attractive in a high rate system which is lightly loaded at most times. It requires that the actuators be connected in series rather than in parallel and that they be equal area actuators (double-ended piston rods). It also requires that the actuators be controlled by bypassing a controlled amount of flow around them. A system of this type offers natural load sharing among the legs in addition to good efficiency. It was, therefore, extensively studied. It was decided to split the drive actuator circuit into two separate circuits, one for each side of the machine. This had the advantages of allowing the flow rates for the two sides to be different when turning and of halving the number of actuators connected in series.

A significant problem in this arrangement is that of returning the legs. The flow to the actuator can be easily reversed by a control valve. The problem is that the legs must always be returned at a rate faster than that used during the contact phase. Since maximum leg speed is governed by the supply flow rate, it is necessary to have the pump output exceed that needed to produce the desired leg return velocity. This is not a major problem when cruising in a tripod gait since the leg return speed for that gait is only slightly greater than the contact phase speed. However, when using high duty factor gaits, leg return speeds may be more than five times contact phase speed. This implies huge flows through the bypass valves of the contact phase legs and consequent large energy losses, particularly since high duty factor gaits are most likely to be used on steep grades where the actuators are heavily loaded.

An additional problem with the series-bypass type of circuit is that it does not lend itself well to position control. In difficult conditions it is expected that a precision footing mode will be used in which the operator will control foot position directly. The bypass valve configuration alone will not zero velocity under no load. It is necessary to use a control valve in series with the actuator and bypass to provide a capability for reversing flow and zeroing velocity. This arrangement introduces nonlinearities into the position control loop and requires that only one leg be moved at a time.

As can be seen, there were very substantial problems in the design of a valve-controlled actuation system. For this reason a careful re-evaluation of the hydrostatic system alternative was initiated. A suitable set of components for such a system was identified, and weight, performance and cost comparisons were carried out. The system studied is actually a two stage-system

with a valve-controlled pilot stage. The circuit is shown schematically in
Figure 5. A small rotary actuator controlled by a servo-valve is used to
control pump swash-plate angle. Surprisingly, the hydrostatic system turned
out to be lighter in weight, competitive in performance and lower in cost.
The weight of the valve-controlled system is driven up by the need for large
reservoirs and oil coolers. The heat generated by operation at unfavorable
actuator conditions must be removed from the oil. The weight of the extensive
lines and manifolding required is also significant. The cost of the valved
system is driven up by the multiple control valves needed to accommodate
different operating conditions. A servo valve frequently costs considerably
more than a variable displacement pump!

Simulation studies of the hydrostatic system configuration using the CSMP
simulation language indicated that small signal response bandwidths of about
20 Hz could be expected for both the drive and leg lift actuators. About 8 Hz
was predicted for the abduction-adduction actuator. The inertia seen by this
actuator is much higher than that seen by the other actuators. These figures
have since been confirmed by tests of the system. Tested bandwidth to the 3dB
point is 20 Hz for the leg lift actuator, 17 Hz for the drive actuator and 7 Hz
for the abduction-adduction actuator. This dynamic response should be adequate
for system operation. It is actually superior to tested valve-controlled
system response [18], although the hydrostatic test system was optimized to a
higher degree then the valved system.

Figure 6 shows the upper leg with the hydrostatic actuation system. A
drive shaft runs down each side of the machine through the centers of the
abduction-adduction bearings. Two pumps for the leg lift and drive actuators are
mounted in the slider box which is the member to which the abduction-adduction
bearing is mounted. They are driven from the drive shaft by means of gear

belts. The third pump, for the abduction-adduction actuator, is mounted in the body and driven by another gear belt.

The rods of the actuators for the leg lift and drive are mounted to the slide box and the barrels are coupled via trunnions to the slider assemblies. This reduces the length needed for the actuator and slider assembly. The leg lift and drive motions are both actuated by pairs of cylinders mounted on opposite sides of the slider box. This is necessary to avoid placing yaw axis moments on the slider assemblies. All actuators are equal area cylinders. This is essential in a hydrostatic system. The cylinders are fed through bores in the piston rods (Figure 7). This avoids the use of flexible hoses. It also allows the active volume of hydraulic fluid to be minimized which assists dynamic response. The cylinder barrels, pump casings etc. are aluminum to reduce weight. Figure 8 shows a laboratory prototype leg fitted with the hydrostatic actuation system.

The adaptive Suspension Vehicle will be powered by a 900 cc motorcycle engine rated at 70 Kw peak power. This unit weighs 90 Kg. The engine is coupled via a clutch to a 0.25 Kw hour storage flywheel. This unit performs several important functions. Firstly it provides a very large input impedance to the actuation system. This is important since the pumps will place strongly fluctuating load torques on the drive shafts. The flywheel also permits regenerative braking to be used with the pumps acting as motors driven by the actuators. It also allows the motor to operate at near optimum conditions. It will permit a controlled shutdown in the event of a loss of motor power. The flywheel package is being designed and assembled by the University of Wisconsin. It features several unique design features. Overspeed protection is provided by shrinking the rim onto a spoked hub. The interference fit is

designed so that it will be lost at a 20% overspeed causing the rim to disengage from the hub. Containment is provided by the outer portion of the rim itself which is a circumferentially wound glass fiber-epoxy composite. This provides failure containment for the inner metal parts. Relatively little containment is needed against failure of the composite rim since it fails in a progressive and controllable manner by matrix disintegration. Thus, the flywheel enclosure is relatively light. A low vacuum is maintained within the flywheel enclosure.

5. Concluding Remarks

In this article we have attempted to present the reasoning behind the choices of leg and vehicle geometry and power train configuration of the ASV. By the time the article appears, the ASV will be assembled and undergoing initial testing. Outdoor tests are scheduled for the second half of 1984.

The project involves development of new technology in many different areas. Much of this new technology may eventually find its way into more conventional robotic applications.

Acknowledgement

The work reported here was supported by the Defense Advanced Research Projects Agency under Contract MDA-903-82-K-0058

References

1. Anon., <u>Logistical Vehicle Off-Road Mobility</u>, Project TCCO 62-5, U.S. Army Transportation Combat Developments Agency, Fort Eustis, Va., February 1967.

2. Bekker, M.G., <u>Introduction to Terrain-Vehicle Systems</u>, University of Michigan Press, Ann Arbor, Michigan, 1969.

3. Mosher, R.S., "Exploring the Potential of a Quadruped," SAE Paper No. 690191, International Automotive Engineering Conference, Detroit, Michigan, January 1969.

4. McGhee, R.B., "Vehicular Legged Locomotion," to appear in <u>Advances in Automation and Robotics</u>, G.N. Saridis Ed., Jai Press, 19.1.

5. Zuk, D., "3D Optical Terrain Sensor for the 1984 ASV," Workshop Summary, AF/DARPA Robotics Workshop, 21-23 March 1983, Denver, Colorado, pp. 1i-11 to 11-27.

6. Gabrielli, G. and Von Karman, T.H., "What Price Speed," <u>Mechanical Engineering</u>, American Society of Mechanical Engineers, Vol. 72, No. 10, 1950, pp. 775-781.

7. Klein, C.A., Olson, K.W. and Pugh, D.R., "Use of Force and Attitude Sensors for Locomotion of a Legged Vehicle over Irregular Terrain," <u>International Journal of Robotics Research</u>, Vol. 2, No. 2, Summer 1983, pp. 3-17, MIT.

8. Song, S.M., Waldron, K.J. and Kinzel, G.L., "Computer-Aided Geometric Design of Legs for a Walking Vehicle," Proceedings of 8th Applied Mechanisms Conference, St. Louis, Missouri, Sept. 19-21, 1983, pp. 70-1 to 70-7.

9. Waldron, K.J., "Robots and Smart Machines," to appear in Standard Handbook of Machine Design, ed. J.E. Shigley and C.R. Mischke, McGraw-Hill, 1984.

10. Raibert, M.H. and Sutherland, I.E., "Machines that Walk," <u>Scientific American</u>, Vol. 248, No. 2, January, 1983, pp. 44-53.

11. Hirose, S. and Umetani, Y., "The Basic Motion Regulation System for a Quadruped Walking Machine," ASME Paper No. 80-DET-34, Design Engineering Technical Conference, Los Angeles, California, September 1980.

12. Gurfinkel, V.S., et al., "Walking Robot with Supervisory Control," <u>Mechanism and Machine Theory</u>, Vol. 16, 1981, pp. 31-36.

13. Russell, M. "Odex 1: The First Functi 'd," <u>Robotics Age</u>, Vol. 5, No. 5, 1983, pp. 12-18.

14. Elftman, H., "Skeletal and Muscular Systems: Structure and Function," Medical Physics, 1944.
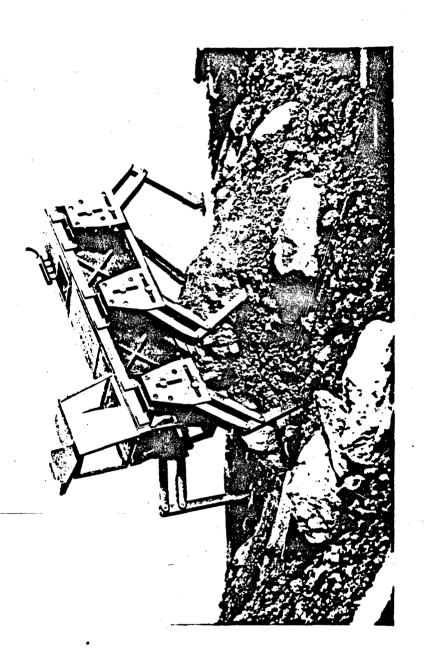
15. Song, S.M., Vohnout, V.J., Waldron, K.J. and Kinzel, G.L., "Computer-Aided Design of a Leg for an Energy Efficient Walking Machine," Proceedings of 7th Applied Mechanisms Conference, Kansas City, Missouri, December 1981, pp. VII-1 to VII-7, to appear in Mechanism and Machine Theory.

16. Kessis, J.J., Rambaut, J.P., Penne, J., "Walking Robot Multi-Level Architecture and Implementation," Proceedings of 4th Symposium on Theory and Practice of Robots and Manipulators, Ed. Morecki, A., Bianchi, G., Kedzior, K., PWN, Polish Scientific Publishers, Warsaw.

17. Vohnout, V.J., Alexander, K.S. and Kinzel, G.L., "The Structural Design of the Legs for a Walking Vehicle," Proceedings of 8th Applied Mechanisms Conference, St. Louis, September 1983, pp. 50-1 to 50-8.

18. Gardner, J.F., Dworak, J.A., Srinivasan, K. and Waldron, K.J., "Design and Testing of a digitally Controlled Hydraulic Actuation System for a Walking Vehicle Leg Mechanism," Proceedings of 8th Applied Mechanisms Conference, St. Louis, September 1983, pp. 2-1 to 2-7.
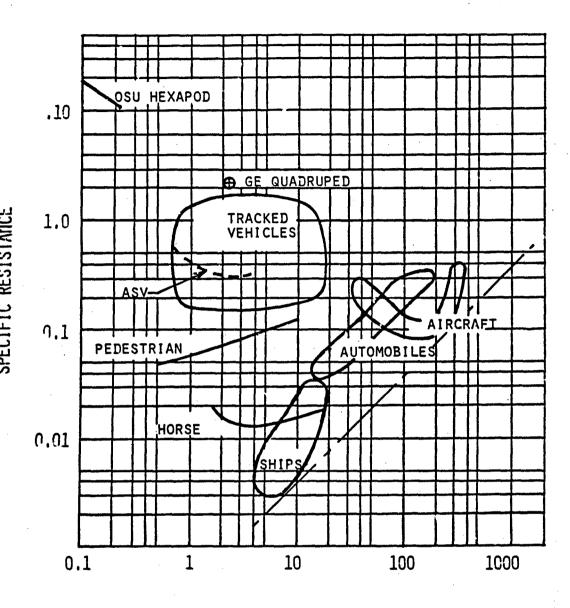
19. Sutherland, I.E., A Walking Robot, The Marcian Chronicles, Pittsburgh, 1982.

Figure Captions

Figure 1:   Concept model of the Adaptive Suspension Vehicle (1/10 scale).
            The box on top of the cab represents the optical terrain scanner.
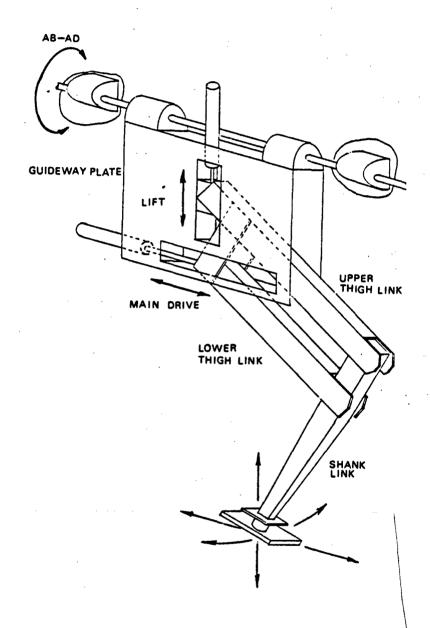
Figure 2:   Specific resistance as a function of speed. Adapted from
            Reference [6] with additional data from References [2] and [3].
            The area for tracked vehicles is enlarged because of sensitivity
            of specific resistance to terrain conditions. The automobile
            data is for prepared surfaces. The dotted curve is the projected
            ASV performance on the basis of laboratory tests of a prototype
            leg.

Figure 3:   Two positions relative to body of a leg with horizontal first
            axis (not shown). In each case work done at one joint is absorbed
            at the other joint.

Figure 4:   Schematic drawing of pantograph leg mechanism.

Figure 5:   Schematic drawing of hydrostatic actuation system.

Figure 6:   Upper leg assembly with hydrostatic actuation system circuit.

Figure 7:   Drive actuator showing feed through rod bore.

Figure 8:   Prototype leg.

Specific resistance versus speed chart showing: OSU HEXAPOD, GE QUADRUPED, TRACKED VEHICLES, ASV, PEDESTRIAN, HORSE, SHIPS, AUTOMOBILES, AIRCRAFT. Y-axis: SPECIFIC RESISTANCE (.10, 1.0, 0.1, 0.01); X-axis: SPEED M/s (0.1, 1, 10, 100, 1000).

AB–AD

GUIDEWAY PLATE

LIFT

UPPER
THIGH LINK

MAIN DRIVE

LOWER
THIGH LINK

SHANK
LINK

LIFT ACTUATORS

SWASHPLATE
CONTROL ACTUATORS

MAIN ACTUATORS

AB/AD ACTUATOR

LOAD SIMULATION CYLINDERS

FOOT VERTICAL "LIFT" CYLINDERS

LIFT HYDROSTATIC PUMP

DRIVE HYDROSTATIC PUMP

FOOT HORIZONTAL "DRIVE" CYLINDERS

LEG LINKS

LEG GUIDEWAY PLATE ASSEMBLIES

CYLINDER END CAP

LOAD ROD

PISTON

CYLINDER

OIL PORT "A"

OIL PORT "B"

OIL INLET ROD

OIL INLET "B"

OIL INLET "A"

CYLINDER END CAP
& TRUNION MOUNT

Appendix 3

Mechanical and Geometric Design of the Adaptive Suspension Vehicle

MECHANICAL AND GEOMETRIC DESIGN OF THE ADAPTIVE SUSPENSION VEHICLE

Kenneth J. Waldron, Shin Min Song,
Shih Liang Wang and Vincent J. Vohnout

Department of Mechanical Engineering
The Ohio State University

Some aspects of the mechanical and geometric design of the Adaptive Suspension Vehicle are presented. In particular, there is an emphasis on aspects of the leg design and vehicle geometry which affect the ability of the vehicle to operate on steep grades or to cross obstacles. A mechanism which maintains the attitude of the foot approximately parallel to the body is described. Geometric aspects of maintaining static stability on steep grades are discussed. Geometric and gait sequence aspects of crossing severe obstacles are also discussed.

1. Introduction

The Adaptive Suspension Vehicle (ASV) will be a mobility system for use in very rough terrain. It uses a legged locomotion principle rather than wheels or tracks. The motivation for using a legged system is that about half of the earth's land surface is inaccessible to wheeled or tracked vehicles [1] but presents little problem to animals using legged locomotion. An artist's conception of the ASV is shown in Figure 1. It's principal characteristics and design goals are summarized in Table 1. The machine will not be a robot since it will carry an operator. However, the operator's role is purely strategic, as in a conventional vehicle.

The ASV will have considerable on-board data processing power and sophisticated environmental sensing capability. This is needed to relieve the operator of the burden of coordinating the eighteen actuated degrees of freedom in the system. In most operating conditions control of leg motion is fully automatic. This type of operation was not feasible prior to the advent of powerful

microcomputers within the last few years. The ASV will be the second, computer coordinated fully self contained walking vehicle when it enters testing in summer 1984. A smaller machine which differs in using a finite state coordination system and which is limited by its mechanical configuration to relatively even terrain, has recently been tested [2]. Further details of the electronic and computational features of the system may be found in the companion paper [3]. The configuration of the sensing system, which is of great importance to operation of the machine, is described in reference [4].

2. Configuration

As can be seen from Figure 1, the vehicle will have six legs arranged in a bilaterally symmetric pattern. The choice of six legs is a compromise between stability and complexity [4,5]. The leg geometry is shown schematically in Figure 2. It is a planar pantograph hinged to the body of the machine about an axis parallel to the longitudinal axis of the body. The reasons for choosing this configuration have been extensively discussed elsewhere [4,6,7]. Its proportions and the dimensions of its working volume are shown in Figure 3. The working volume dimensions are basically fixed by the limits of motion of the actuating slides. These produce a rectangular generating curve. The rectangle is modified by a mechanical limit on knee joint motion, which clips off a top corner, and by geometric limits of the mechanism which clip the bottom corners.

As is shown in Figure 1, the shank of the leg will be cranked. This is necessary to avoid contact of the shank with the ground during some obstacle crossing maneuvers. The leg will also be fitted with a passive hydraulic system which will maintain the sole of the foot approximately parallel to the body at all times. This system is shown schematically in Figure 4. The master cylinder is actuated by rotation of the upper leg members relative to the horizontal slide. The master and slave cylinders have the same diameter so the sum of their displacements is constant. The foot alignment is only approximate but varies by less than 3° over the working volume. An exact system is, of course, possible but entails a more complicated mechanical configuration. A controlled compliance is built into the system allowing some angulation of the foot under load. Excessive angulation will be prevented by mechanical stops. The compliance is provided by accumulators and may be adjusted by altering the pressure maintained in the system when unloaded.

3. Gradability

A major influence on the geometric configuration of the vehicle is the requirement for crossing large obstacles and for operation on steep gradients.

Operation on extreme gradients is potentially limited by three factors. The first of these is vehicle geometry. The second is capability of maintaining adequate traction. The third is the availability of sufficient driving power from the power train.

Vehicle geometry limits the gradient on which the vehicle can operate because of the necessity of maintaining static stability. As is shown in Figure 5a and b, the effect of climbing directly up a slope with the body attitude parallel to the slope is to move the center of mass backward with respect to the support polygon. There are two basic strategies available to counteract this effect. The first of these is to lower the vehicle walking height while maintaining the same body attitude. This reduces the distance from the center of mass to the ground and hence reduces the displacement of the center of mass (Figure 5c). Since the legs are capable of executing a full stroke even when raised to near the top of their working envelopes, this is an effective strategy.

The second strategy is to reduce the inclination of the body by extending the rear legs more than the front. This means that the legs must stroke along an inclined line relative to the body (Figure 5d). For the high ratio of vehicle length to vertical leg lift adopted for the ASV, this is not a very effective strategy. The maximum gradient, which can be handled without reducing the strokes of the front and rear legs can be found from Figure 5d to be:

$$G = \frac{H}{2p + L}$$

where H is the maximum leg lift height, p is the pitch of the legs and L the maximum leg stroke. The gradient G is related to the slope angle, α, shown in Figure 5a by the relationship $G = \tan \alpha$. For the ASV this works out to 25%. This strategy is not really compatible with the strategy of lowering walking height since reducing vehicle body height reduces the available portion of the leg working envelope in the vertical direction. One result of using this strategy is that the foot attitude maintenance mechanism would cause the machine to try to stand on its toes. This might carry a traction advantage in some soil conditions. Otherwise, this strategy is less attractive than the first.

The stability margin can be increased for the lower duty cycle wave gaits [5] by reducing the leg stroke. This is most easily seen for a tripod gait (Figure 6). Considering, for the moment, level walking and referring to Figure 6b, the static stability margin, S, may be expressed as:

$$S = \frac{1}{2} (p-\ell)$$

where p is leg pitch, as before, and ℓ is leg stroke. Further, when the leg stroke is shorter than its maximum (L), the center of the stroke can be moved

backwards relative to the vehicle body a distance

$$d = \frac{1}{2} (L-\ell)$$

This has the effect of moving the entire support polygon backward a distance d relative to the center of mass. Therefore, referring again to Figure 5a and assuming that loss of static stability by tipping backward is the critical condition, the longitudinal static stability margin becomes:

$$S = \frac{1}{2} (p-\ell) + \frac{1}{2} (L-\ell) = vG$$

or

$$S = \frac{1}{2} (L+p) - \ell - vG \qquad (1)$$

The limiting slope is obtained by letting S and ℓ approach 0 and reducing v to its minimum value V.

$$G_{max} = \frac{L + p}{2V}$$

For the ASV, this works out to 240%. Even if one takes $\ell = L/2$ as a working minimum stroke the gradient at which stability is lost is

$$G = \frac{p}{2V}$$

which is 110% for the ASV.

As has been noted many times [5], the stability margins of wave gaits increase with increasing duty factor. Considering only the simultaneous cases, the equation corresponding to equation 1 is

$$S = \frac{1}{2} (L+p) - \frac{5\ell}{8} - vG \qquad (2)$$

for a parallelogram gait and

$$S = \frac{1}{2} (L+p) - \frac{2\ell}{5} - vG \qquad (3)$$

for a pentapod gait. Note that the extreme slopes given by these expressions are the same as that given by equation 1. If one takes $\ell = L/2$ as the working minimum stroke one gets 160% and 190% respectively for ASV dimensions.

Turning now to cross-slope locomotion one must consider both the lateral and longitudinal stability margin. Here it is assumed that the vehicle will move in the longitudinal rather than the lateral direction, although sidestepping might be a useful maneuver on a steep slope.

In these conditions, as shown in Figure 7, reduction of body inclination by extending the down-slope legs is the most effective strategy. The maximum gradient which can be handled with the legs vertical and the body horizontal is

$$G = \frac{H}{W}$$

where W is track width. This is 73% for the ASV. Steeper slopes may be handled

without loss of longitudinal stability margin by adducting the legs as shown in Figure 7b. This, of course, entails a loss of lateral stability margin. Alternatively, the body may be allowed to tilt as shown in Figure 7c. This increases the lateral stability margin on the down-slope side but reduces that on the up-slope side. It also reduces the longitudinal stability margin. This effect is illustrated for a tripod gait in Figure 7d.

The matter of traction is much less clear-cut and depends strongly on the nature of the soil over which the vehicle is moving. In principle, a foot which descends vertically, and which gains in traction with deformation of the soil should enjoy an advantage over a wheel, which loses traction with soil deformation, over a considerable range of soil conditions [8]. Obviously, use of differently designed soles for different soils offers considerable advantages. Biological legged systems can maintain traction on loose soil slopes quite close to the natural angle of repose which, of course, represents the upper limit for any traction system. It remains to be seen how much cross-slope traction will be affected by the lack of ankle accommodation to side slopes. It is probable that traction limitations will be the true determinant of maneuverability on extreme slopes.

As far as the availability of power is concerned, if a gross weight of 2650 kg is assumed, at the designed cruise speed of 2.25 m/s and the specified gradient of 60%, a power of 30 kW is required to overcome gravity. Since it is projected that 18 kW will be needed to drive the vehicle on level ground at this speed, the total power needed to climb this grade is 48 kW. This is less than the rated peak power of the engine which is 67 kW. In fact it will be possible to draw even higher power for a limited period of time because of the use of an energy storage flywheel. Thus, grade climbing ability should not be limited by available power.

4. Obstacle Crossing

The vehicle geometry is also strongly affected by the requirements of obstacle crossing. Figure 8 shows the critical positions for: (a) ditch crossing, and (b) step climbing. The vehicle will be able to cross a 2.15 m ditch or a 1.65 m vertical step.

The sequencing of leg movements for crossing obstacles has been extensively investigated both by means of computer simulation, using an Evans and Sutherland PS 300 display system, and by means of high speed photographs of insects crossing obstacles of similar geometry [9]. The appropriate sequence of movements turns out to be remarkably similar regardless of the nature of the obstacle. It is illustrated in Figure 9 for ditch crossing. Starting with all legs in

their mid-position and with the front feet at the edge of the obstacle, the rear legs are first brought as far forward as possible with the center of mass of the body moving forward to a position between the middle and front legs. At the same time the center of mass is brought forward. The middle legs are then brought forward in a paired movement and the feet are placed alongside the front feet. The front legs are then lifted and extended across the obstacle and placed on the far side while the center of mass moves to a position just behind the middle feet. The center of mass is moved forward of the middle feet and the rear legs are lifted and brought forward to place the rear feet alongside the middle feet. The middle legs are then lifted and the feet are placed across the obstacle. The front feet are then moved forward with the center of mass again moving to a position just behind the middle feet. The center of mass is moved forward of the middle feet and the rear legs are lifted and moved across the obstacle. The middle and front legs are then moved forward to resume stance with the legs centered.

Notice that all leg movements are paired with the two legs on opposite sides of the machine moving together. Notice also that the movements can be characterized as three cycles of four similar movements: (i) movement of the body to bring the center of mass forward of the middle feet, (ii) movement of the rear legs accompanied by further body movement, (iii) movement of the middle legs, (iv) movement of the front legs again with accompanying body movement. The paired movements allow the center of mass to be placed as close as possible to the obstacle without losing stability. The rear to front cycle is similar to the rear to front cycles found in optimally stable wave gaits for walking on even terrain. Movements in the vertical direction and about the pitch axis are superimposed on those above for crossing obstacles such as vertical steps and walls. In fact, a slightly wider ditch can be crossed by adding pitch movements. The basic motion sequence remains the same in all cases.

5. Summary

The Adaptive Suspension Vehicle is now under construction. A full scale leg prototype has been tested in the laboratory and has demonstrated the effectiveness of the hydrostatic drive concept [4]. The vehicle is scheduled for testing in summer 1984. Apart from the aspects of the design discussed here, details of structural design may be found in Reference [10].

Acknowledgement

References

1. Anon., <u>Logistical Vehicle Off-Road Mobility</u>, Project TCCO 62-5, U.S. Army Transportation Combat Developments Agency, Fort Eustis, Va., February 1967.

2. Raibert, M.H. and Sutherland, I.E., "Machines that Walk," <u>Scientific American</u>, Vol. 248, No. 2, January, 1983, pp. 44-53.

3. McGhee, R.B., Orin, D.E., Pugh, D.R. and Patterson, M.R., "A Hierarchically-Structured System for Computer Control of a Hexapod Walking Vehicle," Fifth CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, June 1984.

4. Waldron, K.J., Vohnout, V.J., Pery, A. and McGhee, R.B., "Configuration Design of the Adaptive Suspension Vehicle," to appear in <u>Robotics Research</u>, Spring 1934.

5. McGhee, R.B., "Vehicular Legged Locomotion," to appear in <u>Advances in Automation and Robotics</u>, G.N. Saridis Ed., Jai Press, 1984.

6. Song, S.M., Waldron, K.J. and Kinzel, G.L., "Computer-Aided Geometric Design of Legs for a Walking Vehicle," Proceedings of 8th Applied Mechanisms Conference, St. Louis, Missouri, Sept. 19-21, 1983, pp. 70-1 to 70-7.

7. Song, S.M., Vohnout, V.J., Waldron, K.J. and Kinzel, G.L., "Computer-Aided Design of a Leg for an Energy Efficient Walking Machine," Proceedings of 7th Applied Mechanisms Conference, Kansas City, Missouri, December 1981, pp. VII-1 to VII-7, to appear in <u>Mechanism and Machine Theory</u>.

8. Bekker, M.G., <u>Introduction to Terrain-Vehicle Systems</u>, University of Michigan Press, Ann Arbor, Michigan, 1969.

9. Pearson, K.G. <u>Cinematographic Analysis of Animal Walking</u>, final report on contract RF 714250-01, University of Alberta, Edmonton, Alberta, Canada, September 30, 1982.

10. Vohnout, V.J., Alexander, K.S. and Kinzel, G L., "The Structural Design of the Legs for a Walking Vehicle," Proceedings of 8th Applied Mechanisms Conference, St. Louis, September 1983, pp. 50-1 to 50-8.

TABLE 1

ASV CHARACTERISTICS	
Dimensions:	5.0 m Long, 2.1 m Wide, 3.3 m High (average), 1.6 m Track
Weight:	260 kg
Payload:	225 kg
Endurance:	10 hrs.
Speed:	2.25 m/s Cruise, 3.6 m/s Dash
Grade Climbing Ability:	>60%, 70% Cross-slope
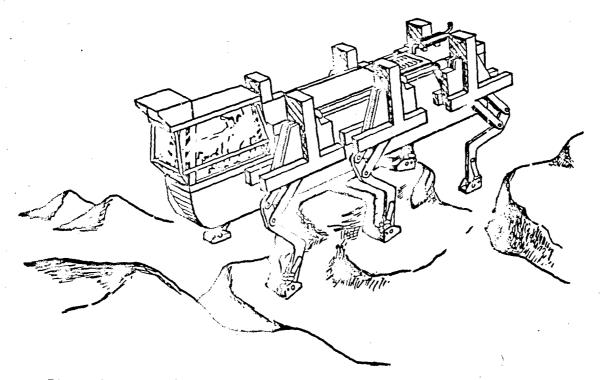
Figure 1: Artist's conception of the Adaptive Suspension Vehicle.
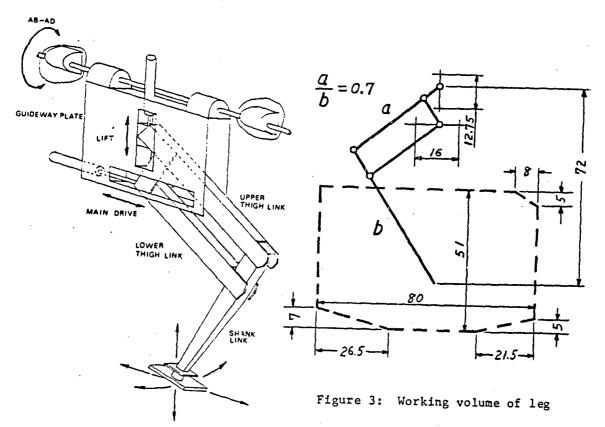
Figure 2: Schematic representation of pantograph leg arrangement

$$\frac{a}{b} = 0.7$$

Figure 3: Working volume of leg
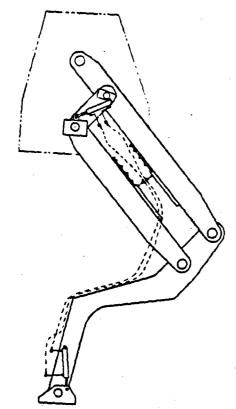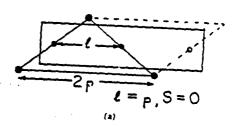
Figure 4:

Schematic representation of foot

attitude maintenance system

$\ell = P, S = 0$

(a)

(b)

(c)
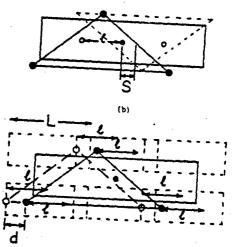
Figure 6:   Effect on stability margin

of shortening stroke

(a)

(b)

(c)

(d)
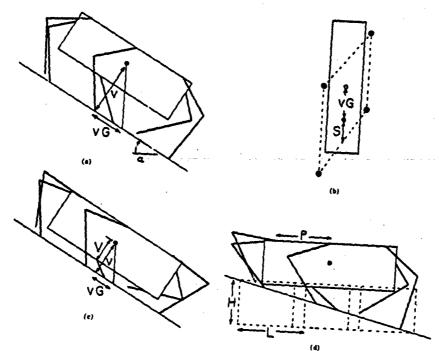
Figure 5:   Maintenance of static stability on gradients
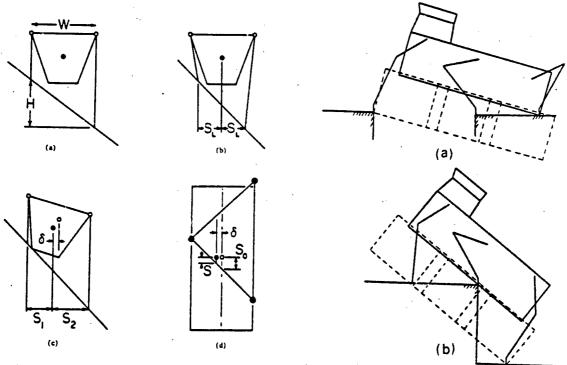
Figure 7: Maintenance of static
stability on cross gradients

Figure 8:

Critical positions in obstacle crossing:

(a) Ditch, (b) Vertical step
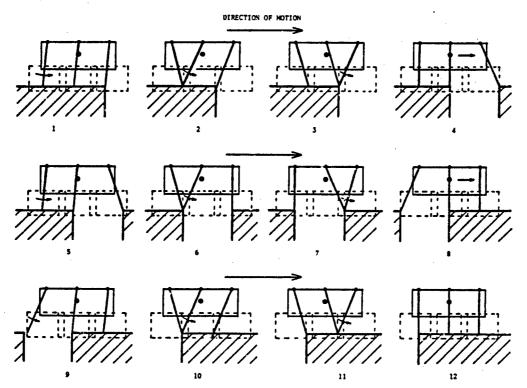
Figure 9: Movement cycles in crossing maximal width ditch

Appendix 4

A Hierarchically-Structured System for Computer Control

of a Hexapod Walking Machine

A HIERARCHICALLY-STRUCTURED SYSTEM FOR

COMPUTER CONTROL OF A HEXAPOD WALKINC MACHINE

R.B. McGhee, D.E. Orin, D.R. Pugh and M.K. Patterson

The Ohio State University
Department of Electrical Engineering
Columbus, Ohio 43210, U.S.A.

SUMMARY

Control of walking machines involves a hierarchy of computational tasks which is naturally suited to a multi-computer implementation. This paper describes the hardware and software for one such system presently under construction. Particular attention is paid to problems of man-machine and intercomputer communication during vehicle operation and to integration of artificial sensory information into higher levels of control.

INTRODUCTION

Designers of off-road vehicles have long been aware that the performance of such machines is inferior to that of cursorial animals in rough-terrain locomotion. The advantages of natural systems arise from intrinsic characteristics of legged locomotion that include greater mobility, higher speed, and reduced energy requirements in comparison to conventional wheeled or tracked automotive vehicles [1]. Research efforts over the past two decades have shown that it is possible to obtain animal-like behavior in machines, provided that the relationship of the supporting elements of the machine to the terrain is actively adjustable to permit effective accommodation of terrain irregularities. Such machines are called "adaptive-suspension vehicles" or, sometimes, simply "walking machines" [1,2].

Early experiments with adaptive-suspension vehicles required direct coordination of individual joint motions by a human operator [2,3]. Although this technique was successful to the extent that excellent off-road mobility was demonstrated by a quadruped walking machine, both in dense obstacle fields and over very soft soil, it was found to be impractical due to the high demands placed on the operator's motion coordination skills, resulting in rapid fatigue and marginal stability. As a result, beginning in about 1970, research was initiated on the possibility of assigning low-level coordination tasks to a computer while requiring the operator to deal with only the more complex vehicle control problems such as route selection and control of speed and heading. These research efforts resulted in successful demonstration of such "supervisory" control in 1977 by two laboratory-scale hexapod walking machines. One of these machines, the OSU Hexapod [4,5], was controlled by

1

a uniprocessor digital computer, while the other, the MGU Hexapod [6], was controlled by a hybrid analog-digital computer. Both were externally powered, and each was connected to its computer by a trailing umbilical cord.

With the advent of modern 16-bit microcomputers, it has become feasible to construct self-contained walking machines with supervisory control realized by an on-board computer. The first such machine walked in late 1982 at Carnegie-Mellon University [7], and was controlled by a uniprocessor computer, the Motorola 68000. Preliminary experiments indicate, however, that refined coordination of motion for rough-terrain locomotion is difficult for a computer of this size. Rather, at the present time, a hierarchical decomposition of the control task for multiprocessor implementation seems to be more appropriate [8,9]. Such an implementation has the further advantage that higher levels of sensing and control, such as the use of computer vision in motion planning and foothold selection, can be added in a modular fashion without significantly altering previous layers of control computer hardware and software [10]. An experiment of this sort is now under way at Ohio State University with respect to a large hydraulically powered hexapod walking machine called the "ASV-84." The mechanical design of this machine is described elsewhere in these proceedings [11]. The purpose of the present paper is to explain the organization of the hardware and software employed in the on-board multiprocessor computer used for control of this vehicle. In what follows, particular attention is paid to problems of man-machine and intercomputer communication, as well as to integration of artificial sensory information into higher levels of control.

2

OPERATIONAL MODES

The ASV-84 is being developed as a test-bed vehicle for the study of sensors, for the evaluation of alternative prime movers and energy distribution systems, and for the further development of control techniques for semi-autonomous or fully autonomous operation. Because the current state of control and sensing technology does not permit fully autonomous operation of walking machines under all terrain conditions, this vehicle is provided with a cab for a human operator who participates in vehicle control at various levels depending upon task and terrain complexity [12]. The operator's cab contains aircraft-style controls and displays, including a three-axis joystick for vehicle steering and speed control. An optical radar mounted on top of the cab provides a dense range map of the terrain immediately ahead of the vehicle which is used in some control modes for foothold selection and body orientation. The following paragraphs provide a brief description of each major control mode envisaged for this vehicle.

Utility

This mode is intended to perform a "pre-flight" checkout to verify correct functioning of all major vehicle subsystems. Both visual observations and redundant transducer signals can be used for this purpose. Self-test software includes some capability to isolate faulty electronic or mechanical modules to facilitate field or shop repair. This mode is also used for reprogramming the control computer.

3

Precision Footing

In this mode, the operator is able to control individual legs by means of joystick, keyboard, or other commands. Control of foot positions is in body-fixed Cartesian coordinates. A CRT display of vehicle stability margin derived from a vertical gyroscope and foot force and position transducers is provided to assist the operator. Feedback of swing-phase foot position relative to terrain is primarily through operator vision, although this function can be assisted by the use of proximity sensors mounted on the vehicle legs. Body motion control can be either automatic [10] or manual in this mode. In the manual mode, the operator can assign his joystick axes either to translational or rotational body velocity control.

Close Maneuvering

This is a three-axis control mode in which the turning center for body rotation can be placed anywhere along the vehicle longitudinal axis. Once this has been done, arbitrary combinations of yaw rotational rate, forward velocity, and lateral velocity can be commanded by a three-axis joystick. Stepping is regulated automatically with swing-phase foot elevation controlled primarily by proximity sensors in order to achieve an operator-specified ground clearance [13]. Body roll and pitch can be regulated either to a fixed set-point or adjusted automatically to conform to the terrain slope [14]. Force feedback is used to control foot loading and to prevent the development of unwanted antagonistic forces between supporting legs [5]. The optical radar is not used in this mode since its field of view is limited to ± 40 degrees in azimuth from the vehicle centerline.

Terrain-Following

All vehicle sensors are used in this mode. The optical radar provides terrain preview data for use by the on-board computer in the selection of foothold locations and the determination of average terrain slope and elevation for body attitude and altitude regulation [15]. During the swing phase of limb motion, proximity sensors on each leg are used in conjunction with optical radar data for local control of foot elevation and for collision avoidance. During support phase, force feedback is used to achieve active compliance to smooth body motion and control leg loading [5,16]. Body turn rate, forward velocity, and lateral velocity are determined by operator inputs [4]. This mode makes use of algorithms for free (non-periodic) gaits in order to improve rough terrain mobility at moderate speeds [12,15,17]. Both the maximum body crab angle [4] and the minimum turning radius are limited to values which ensure that all footholds used by the vehicle lie on terrain which has been mapped by the forward-looking optical radar.

Cruise

This is the most efficient mode of operation with respect to fuel economy. It is suitable for locomotion over reasonably smooth terrain. The design goal for vehicle speed in this mode is 8 km/hr. Body crab angle is limited to a relatively small value and minimum turning radius is of the order of several body lengths. The on-board computer uses optical radar data to determine a desired foot-lift height during the swing phase of leg motion. The proximity sensors assist in control of foot velocity at ground impact. It is expected that cruise mode will utilize an "equiphase" tripod gait in which the footfalls of supporting legs are evenly spaced in time in order to

5

minimize shock and vibration transmitted to the vehicle body by the cyclic action of the legs [18].

Dash

This is a projected mode in which all aspects of vehicle performance will be sacrificed for speed. Specifically, maneuverability will be limited, ride characteristics are expected to be rough, and fuel economy is likely to be poor. An alternating tripod gait [2] will probably be used for this purpose. The control computer will command body attitude and swing-phase foot-lift height. Top speed should be of the order of 12 km/hr in this mode.

COMPUTER ARCHITECTURE

Experience to date with the OSU Hexapod indicates that a network of microcomputers probably provides the most effective configuration for the ASV-84 on-board computer [8]. The following paragraphs describe one partitioning of computational tasks among such computers which is consistent with the above defined control modes and which tends to minimize intercomputer data communication rates. Figure 1 is a graphical representation of the computer architecture implied by this partitioning. All computers in this diagram are physically realized in terms of Intel iSBC 86/30 single-board computers. These computers make use of the 8086 microprocessor with an optional incorporation of the 8087 floating-point coprocessor as well as 16 channels of analog input and 8 channels of analog output lines. A description of the function of each computer in this system follows:

6

Guidance Computer

This computer receives range data from the optical radar and produces a stabilized terrain map in earth-fixed Cartesian coordinates at a 2 Hz frame rate. As shown in Figure 1, it also receives information from a coordination computer. This information consists of body attitude, body altitude, all six body velocity components relative to the supporting terrain, foot support states, individual foot positions expressed in body coordinates, and operator velocity and steering requests. The output of this computer consists of timing and trajectory commands for swing-phase legs and of commanded body positions and velocities. Eventually, the guidance computer should also control the scanning angles of the optical radar, although this is not contemplated for the initial phases of ASV-84 testing. The guidance computer contains four iSBC 86/30 boards [15].

Coordination Computer

This computer is the master computer in the sense that all intercomputer communications must pass through it. Specifically, the coordination computer receives designated footholds, swing-phase foot trajectories, and commanded body positions and velocities from the guidance computer. It also receives operator commands from a cockpit computer as well as body attitude information from an inertial measurement unit. It then communicates with leg control computers in order to provide appropriate foot motion commands. These commands must be derived as a compromise between operator inputs and constraints imposed by the dynamic and kinematic limitations of individual legs. As complicated as this process sounds, it is well understood with respect to control of the OSU Hexapod Vehicle [4,5,8,10]. For legs in support phase, commands from the coordination computer consist of desired foot

velocities and forces in body-fixed Cartesian coordinates. For swing-phase legs, either point-to-point control or continuous velocity control is used, depending upon the particular operational mode. The coordination computer is composed of two iSBC 86/30 boards functioning in a partial duplex mode in which each computer monitors the other's function. Both of these computers are connected directly to primary system sensors and either can automatically initiate emergency shut-down procedures in case of failure of a major vehicle subsystem.

Cockpit Computer

This computer reads levers, dials, and push buttons manipulated by the human operator. It also operates all cockpit instrumentation, including at least one CRT display. This display is used during normal operation as well as for system checkout and diagnostics.

Leg Control Computers

The actuator control valves for each leg are operated by a control computer associated with that leg. Each such computer receives commands from the coordination computer in body-fixed coordinates. It then translates these to joint coordinates and implements the resulting motions through closed-loop velocity control of each joint. Jacobian control, as used in the OSU Hexapod, permits this to be done without explicitly solving for desired leg joint angles [16]. In certain operational modes, leg computers use proximity sensor information to maintain foot clearance above the terrain during the swing phase of leg motion and also use this information to achieve a soft landing at the end of swing phase.

8

COMPUTER COMMUNICATIONS

Two different approaches to intercomputer communications are used in the ASV-84 computing system. For the guidance computer, data rates are low enough to permit communication with the coordination computer by means of parallel data ports. Leg control computers involve higher data rates, and will therefore be connected with the coordination computer through a high-speed data bus (Intel Multibus), using a shared memory "mailbox" type of communication. The same communication technique is used for the cockpit computer. Transmission of partial data blocks between asynchronous processes is prevented in all cases by a buffering scheme based on dynamic reassignment of input and output buffers for each communication channel.

SUMMARY AND CONCLUSIONS

As of the time of this writing, a breadboard version of the ASV-84 computer has been completed. The optical radar system has been delivered and laboratory testing with the guidance computer is under way. Software for the precision footing mode of control has been completed and validated using the OSU Hexapod as a physical test-bed. Terrain-following software has been installed on the breadboard computer and is being evaluated using a PDP-11/70 computer to simulate the optical radar, terrain, and vehicle mechanical system. Outdoor testing of the completed ASV-84 vehicle is scheduled for late 1984. The authors hope that this testing will establish a new level of performance for walking machines and will facilitate the development of specialized adaptive-suspension vehicles designed for specific applications.

REFERENCES

1.   Waldron, K.J., et al., "Configuration Design of the Adaptive-Suspension Vehicle," _International Journal of Robotics Research_, Vol. 3, No. 2, April, 1984.

2.   McGhee, R.B., "Vehicular Legged Locomotion," in _Advances in Robotics and Automation_, Vol. 1, ed. by G.N. Saridis, Jai Press, Inc., Greenwich, Conn., 1984.

3.   Mosher, R.S., "Exploring the Potential of a Quadruped," SAE Paper No. 690191, International Automotive Engineering Conference, Detroit, MI, January, 1969.

4.   Orin, D.E., "Supervisory Control of a Multilegged Robot," _International Journal of Robotics Research_, Vol. 1, No. 1, Spring, 1982, pp. 79-91.

5.   Klein, C.A., Olson, K.W., and Pugh, D.R., "Use of Force and Attitude Sensors for Locomotion of a Legged Vehicle over Irregular Terrain," _International Journal of Robotics Research_, Vol. 2, No. 2, 1983, pp. 3-17.

6.   Gurfinkel, V.S., et al., "Walking Robot with Supervisory Control," _Mechanism and Machine Theory_, Vol. 16, 1981, pp. 31-36.

7.   Raibert, M.H., and Sutherland, I.E., "Machines that Walk," _Scientific American_, Vol. 248, No. 2, January, 1983, pp. 44-53.

8.   Klein, C.A., and Wahawisan, W., "Use of a Multiprocessor for Control of a Robotic System," _International Journal of Robotics Research_, Vol. 1, No. 2, Summer, 1982, pp. 45-59.

9.   Russell, M., "Odex 1: The First Functionoid," _Robotics Age_, Vol. 5, No. 5, 1983, pp. 12-18.

10.   Ozguner, F., Tsai, S.J., and McGhee, R.B., "An Approach to the Use of Terrain Preview Information in Rough-Terrain Locomotion by a Hexapod Walking Machine," _International Journal of Robotics Research_, Vol. 3, No. 2, April, 1984.

11.   Waldron, K.J., et al., "Mechanical and Geometric Design of the Adaptive-Suspension Vehicle," _Proc. of ROMANSY-84 Symposium_, Udine, Italy, June, 1984.

12.   Klein, C.A., and Patterson, M.R., "Computer Coordination of Limb Motion for Locomotion of a Multiple-Armed Robot for Space Assembly," _IEEE Trans. on Systems, Man, and Cybernetics_, Vol. SMC-12, No. 6, December, 1982, pp. 913-919.

13.   Broerman, K.R., _Development of a Proximity Sensor System for Foot Altitude Control of a Terrain-Adaptive Hexapod Robot_, M.S. thesis, The Ohio State University, Columbus, OH 43210, August, 1983.

14.   Lee, W. J., *A Simulation Study of Interactive Computer Control for Rough-Terrain Locomotion by a Hexapod Walking Machine*, Ph.D. dissertation, The Ohio State University, Columbus, OH 43210, March, 1984.

15.   Patterson, M.R., Reidy, J.J., and Brownstein, B.J., *Guidance and Actuation Techniques for an Adaptively Controlled Vehicle*, Final Technical Report, Battelle Columbus Laboratories, Columbus, OH 43201, March, 1983.

16.   Klein, C.A., and Briggs, R.L., "Use of Active Compliance in the Control of Legged Vehicles," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 7, July, 1980, pp. 393-400.

17.   McGhee, R.B., and Iswandhi, G.I., "Adaptive Locomotion of a Multilegged Robot over Rough Terrain," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-9, No. 4, April, 1979, pp. 176-182.

18.   Wang, S.L., *The Study of a Hexapod Walking Vehicle's Maneuverability over Level Ground and Obstacles and Its Computer Simulation*, M.S. thesis, The Ohio State University, Columbus, OH 43210, August, 1983.

Communication Technique

① : Parallel ports

② : Analog data lines

③ : Intel Multibus

INERTIAL MEASUREMENT UNIT

COCKPIT DISPLAYS AND CONTROLS

COCKPIT COMPUTER

COORDINATION COMPUTER

OPTICAL RADAR SYSTEM

GUIDANCE COMPUTER

LEG CONTROL COMPUTER

HYDRAULIC VALVES

LEG SENSORS

LEG CONTROL COMPUTER

HYDRAULIC VALVES

LEG SENSORS

(Six Units)
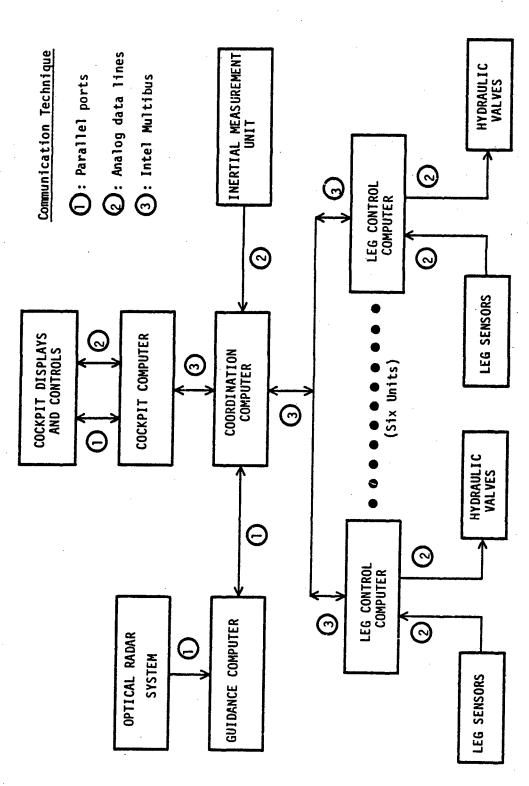
Figure 1. On-Board Computer Architecture for ASV-84 Vehicle.

Appendix 5

Development of a Proximity Sensor System for
Foot Altitude Control of a Terrain-Adaptive Hexapod Robot

DEVELOPMENT OF A PROXIMITY SENSOR SYSTEM FOR FOOT

ALTITUDE CONTROL OF A TERRAIN-ADAPTIVE HEXAPOD ROBOT

A Thesis

Presented in Partial Fulfillment of the Requirements
for the Degree Master of Science

by

Keith Robert Broerman, B.S.E.E.

The Ohio State University
1983

Approved by

Kail W. Olson
Advisor
Department of Electrical Engineering

to my parents

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

x

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background

A deaf, dumb, and blind man without the sense of touch or smell is incapable of interacting with his environment. When his local situation is not known _a priori_, the possession of a single sense, that of sight, can greatly increase his level of environmental interaction. In the same way, a robot without any external sensors can effectively operate only within a completely determinate environment. Thus, the local adaptability of both the man and the robot can be improved by providing them with proximity sensors. Such sensors are used to roughly simulate the sense of sight.

As might be expected, the development of proximity sensing aids for the blind has preceeded the development of sensors for use in robotics applications [1,2,3]. Whatever its origin, the objective of any proximity sensor is the same: to mimic the sense of sight by determining the distance of objects from the sensor being monitored. This distance or range information can then be used in some type of real-time feedback control scheme. One example of feedback, or closed-loop, control would be that of a robot placed alongside an

assembly line in which the robot is programmed to pick up moving objects and then place them into a hopper. A proximity sensor would be used to locate the object's position prior to its removal by the robot.

In this work a proximity sensor system has been developed for use on the OSU Hexapod Robot. The Hexapod, constructed by researchers at The Ohio State University, was developed to study the problems associated with the computer control of multi-jointed vehicles. It possesses six legs with three independently-powered joints per leg, for a total of eighteen degrees of freedom. See Figure 1.1. The Hexapod is presently controlled by an off-board DEC PDP 11/70 minicomputer through a digital data link umbilical cord. Prior to this work the Hexapod satisfactorily demonstrated several modes of locomotion [4], and the addition of a vertical gyro and vector force sensors [5] enabled the vehicle to traverse irregular terrain while maintaining a level body orientation. A review of the existing electrical and mechanical structure of the OSU Hexapod can be found in Pugh [6], Briggs [7], and Buckett [8].

The objective of this thesis is to develop a proximity sensor system to gage Hexapod foot altitudes and to appropriately control the position and velocity of each leg while in its transfer phase of motion, especially during foot descent.

Prior to this work, the only method of estimating Hexapod foot altitudes was to convert known joint positions into foot altitudes using a direct kinematics approach, given the vehicle's body height. Although the calculation of foot altitude based upon these parameters is straightforward, the modification of leg velocity during foot descent requires a priori knowledge of the terrain being traversed.

2

Figure 1.1

The OSU Hexapod.

3

Of course, terrain information is not required if foot-altitude sensors are available for each leg.

Due to a low level of research priority and lack of a readily available foot-altitude sensing device, leg velocity modification based upon foot altitude was not previously implemented on the OSU Hexapod. This resulted in a "stomping" effect while walking, even while on level terrain. Due to the high velocity of each leg at ground impact, both the drive motors and the vehicle frame were subjected to undue amounts of physical stress.

The use of proximity sensors in this thesis should thus give the Hexapod a much smoother stride and an ability to sense terrain depressions or obstacles located in the vicinity of each foot.

1.2 Organization

Chapter 2 provides a survey of proximity sensing techniques that are applicable to the field of robotics. The development of a 40 kHz ultrasonic rangefinding system is discussed in Chapter 3. Experimental results of the system are given in Chapter 4. Application of the sensor system to the OSU Hexapod is described in Chapter 5. In Chapter 6, experimental results of the integrated sensor system are given. Chapter 7 summarizes the results obtained from this research and suggests areas where further work is needed.

4

A SURVEY OF PROXIMITY SENSING TECHNIQUES

In this chapter a brief survey of proximity sensing techniques that can be applied to the field of robotics is given. The term 'proximity sensing' as used in this work will generally refer to non-contact object detection, and where possible, sensor-to-object distance measurement. Several sensing techniques presented below provide only an indication of an object's presence or absence. This binary "go/no-go" sensing can be implemented in several ways: with tactile sensing, capacitive or inductive sensing, pneumatic sensing, or optical sensing. Actual sensor-to-object distance measurement is also discussed and can be implemented with either high-frequency sound or electromagnetic radiation.

2.1 Tactile Sensing

Tactile sensing requires direct physical contact with an object in order to detect its presence. Hence, tactile sensing is not, strictly speaking, a proximity sensing technique. However, it is mentioned frequently in the literature and so will be included here for the sake of completeness.

5

Tactile, or touch, sensing is used quite often in assembly line-oriented processes and is usually implemented with simple contact or limit switches. Such switches, for example, are used in material handling operations to indicate the position and/or orientation of transported workpieces. An interesting use of contact switches as applied to robotics can be found in Umetani [9]. In this example a multi-linked active-cord mechanism is equipped with an array of forty such switches to detect the presence of guide walls.

Tactile sensing by definition requires direct physical contact with the object and hence the sensing device is subject to mechanical wear. This problem may be resolved by using non-contacting sensing techniques which include the inductive, capacitive, pneumatic, and optical sensing methods.

2.2 Inductive Sensing

Inductive sensing requires the use of a magnetic field to determine the presence of an object or target. This presence is typically indicated by an increase in field coil inductance as the target is placed in or near the field. Two major disadvantages of this method are that the target material must be metallic and that the maximum sensing range is approximately one half that of the sensor diameter. A metal detector is a common application of the inductive sensing technique [10].

2.3 Capacitive Sensing

Capacitive sensing is similar to inductive sensing in that the maximum range is also dependent upon the sensor diameter. An exposed

6

capacitor is typically used as a positive feedback element in an amplifier that is biased at the edge of oscillation. As an object approaches the capacitor, the amplifier is forced into oscillation, due to the increase in sensor capacitance. It is this oscillation which provides the presence/absence signal. In principle, the capacitive sensor will respond to all target materials, but in practice this system is difficult to use [10].

Capacitive touch-plate switches are a common app'ication of the capacitive sensing technique. Such switches operate on finger contact and are found in elevators and other high-use areas where their inherent reliability (no moving parts) and long life are major considerations in their selection.

2.4 Pneumatic Sensing

Pneumatic sensing requires the use of pressurized air to make the binary judgement of an object's presence or absence. Pneumatic sensors are generally very rugged and usually have no moving parts. They typically operate with a sensor-to-target separation distance that is cn the order of millimeters. These sensors are usually implemented in one of three ways: as a back-pressure sensor, as an interruptable-jet sensor, or as a cone-jet sensor.

In a back-pressure sensor system the foreign object inhibits the flow of pressurized air from a venting nozzle, which causes the pressure in the venting chamber to rise. This increase in chamber pressure thus signals the presence of an object.

With an interruptable-jet sensor system a supply nozzle and receiver are axially aligned but separated by a gap [11]. When a

7

foreign object enters this gap the air stream is deflected and the receiver pressure is reduced, thus signaling a target. In this arrangement it is not unusual for the air gap to be several centimeters in length.

A third method of pneumatic sensing uses cone jet sensors to determine the presence of a target. With these sensors an output sensing nozzle records a reduction in pressure when the emitted conical air jet is deflected by the foreign object. Such sensors are used to position a robot manipulator hand over circular targets [12]. In this implementation the nozzle-to-target separation distance was fixed at 2 mm.

2.5 Optical Sensing

Optical sensing at millimeter distances generally employ a photoemitter-detector pair configured in one of two ways, as in pneumatic sensing: an axially mounted light source and receiver that requires the presence of an object to break the beam of light, or a side-by-side emitter-detector mounting that uses the object as a reflector. In either case the light source is usually an infrared light-emitting diode (L.E.D.), while the receiver is typically a phototransistor or photodiode. These devices are usually supplied with visible-light filters which reduce the saturation effect that ambient light has upon the receiver. Light-source modulation can also be used to reduce the problems associated with ambient light and noisy environments.

A common example of optical sensing is that of a computer

8
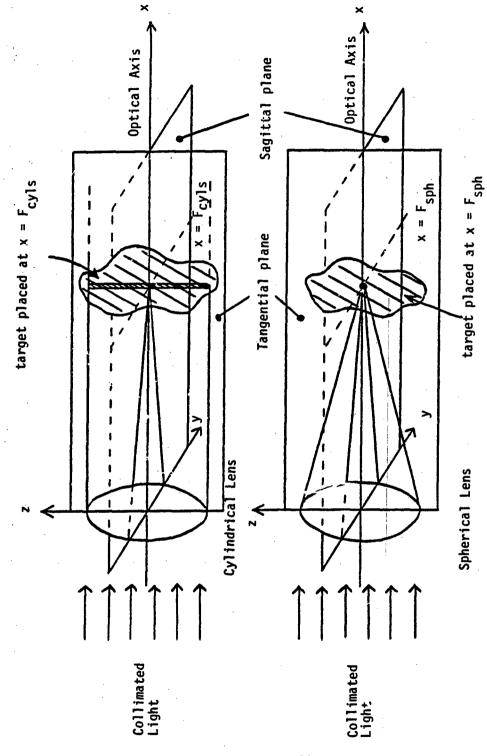
punch-card reader. Arrays of sensors, configured axially as described above, are used to determine the presence or absence of punched holes in the computer cards. In this way data can be quickly scanned and then stored in computer memory. A computer paper-tape reader operates in exactly the same manner.

A novel optical technique which can sense a target's linear displacement employs a collimated light source and a compound lens made from a spherical lens placed in contact with a cylindrical lens. This technique is described below.
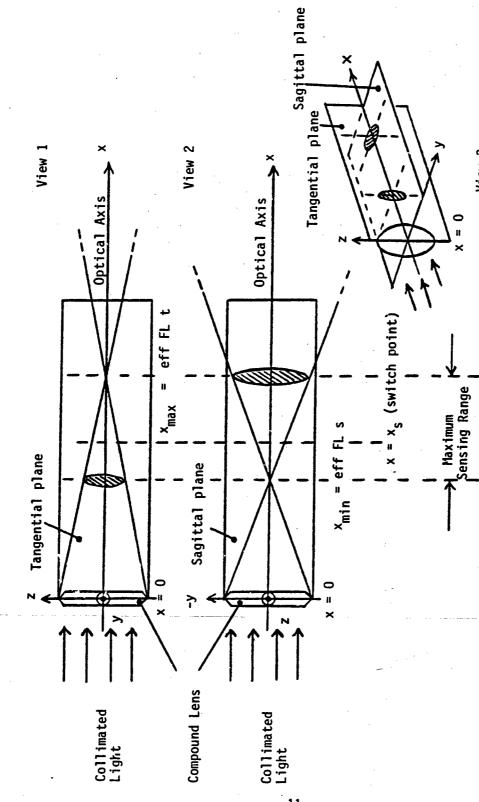
A cylindrical lens differs from a spherical lens in that it has a fixed focal length in only its sagittal plane. See Figure 2.1. Thus, when the lens is illuminated by a collimated light source, and a plane surface is placed perpendicular to the lens' optical axis, a thin strip of light will be observed when the target lies at $x = F_{cyls}$. In this notation F_{cyls} is the focal length of the cylindrical lens in the sagittal plane. The resulting strip of light is due to the fact that the lens has an infinite focal length in the tangential plane; hence, in this plane it has no converging power. A spherical lens, in contrast, has a finite focal length $x = F_{sph}$ that is the same in both the sagittal and tangential planes.

The combination of the spherical and cylindrical lenses thus produces a compound lens whose effective focal length in the sagittal plane, eff FL s, is smaller than the effective focal length in the tangential plane, eff FL t. This difference in effective focal lengths permits us to judge a target's linear position if the target is constrained to lie along the compound lens' optical axis from $x =$ eff FL s to $x =$ eff FL t. See Figure 2.2. The method of detection

9

Figure 2.1

Target Illumination Patterns as a Function of Focusing Lens Type.

10

Figure 2.2

Target Illumination Patterns as a Function of Linear Target Displacement.

11

is as follows: when the target is located at

$$x = x(max) = eff\ FL\ t,\qquad\qquad(2.1)$$

the light source and lens combination illuminates the target with an elliptical pattern whose major axis lies in the sagittal plane. Moving the target toward the compound lens gradually shortens this ellipse until the major and minor axes become equal in length. This "switch point" is a function of the effective focal lengths and is given by

$$x\ switchpoint = x(sw) = \frac{2\ P(eff\ FL)}{S(eff\ FL)}\qquad\qquad(2.2)$$

where P(eff FL) and S(eff FL) represent the product and sum of the effective focal lengths, respectively. As the target range is decreased below this switchpoint the major and minor axes of the illuminated pattern switch, so that at

$$x = x(min) = eff\ FL\ s,\qquad\qquad(2.3)$$

the ellipse's major axis then lies in the tangential plane.

Target position is derived from the ratio of the ellipse's major and minor axes, whose lengths can be sensed with a digital array camera or by an array of discrete photodetectors.

This optical illumination technique may be used for measuring large (greater than one foot) linear target displacements provided

12

that the lenses used have correspondingly large focal lengths. However, due to practical considerations, large focal lengths dictate physically large lenses which may not be available or which may be too large for a practical implementation. At the other extreme, a version of this sensing method is currently used in laser videodisk technology to control the vertical position of an optical pickup head [13]. In this example the typical vertical variation (e.g., due to a warped videodisk) is approximately ± 0.5 μm.

Among the sensing techniques presented heretofore, only the optical illumination scheme of section 2.5 provided a continuous (non-binary) indication of target range. Several other techniqu. ; exist that can provide values of separation distance over a prescribed operating range. These techniques utilize either high-frequency sound waves (ultrasound) or a form of electromagnetic radiation (either infrared or microwave) in determining a value of target range.

2.6 Ultrasonic Ranging

Ultrasound can be used in several ways to determine a target's linear position. Three widely used methods are described below: pulse-echo measurement, phase-delay measurement, and frequency-deviation measurement. In each case an ultrasonic transmitter and receiver assembly is required for target detection.

2.6.1 Pulse-Echo Measurement

In the pulse-echo technique a fixed number of cycles of constant-frequency ultrasound is beamed at a target. Typical ultrasonic frequencies used include 40 kHz and 200 kHz [14]. The transmitted

13

sound waves are then reflected from the target back to the receiver, which waits for the first returning "echo", i.e., the first returning ultrasonic energy having the same frequency as that transmitted.

To calculate target range a timer is used to determine the elapsed time from the transmitted burst to the first received echo. This elapsed, or round-trip, time is then used to calculate target range as follows:

$$\text{target range} = V_{air} \left(\frac{\text{round-trip time}}{2} \right) \tag{2.4}$$

where V_{air} is the velocity of propagation of sound in air and is given by

$$V_{air} = 1087.14 \sqrt{1 + T\ (°C)/273}\ \ \text{ft/sec.} \tag{2.5}$$

Note that the velocity is temperature dependent.

With this pulse-echo technique, a single ultrasonic transducer (usually piezo-electric) can be used in both the transmit and receive modes. Characteristically, these transducers exhibit mechanical ringing after they have been electrically turned off, as the transducer faces are not heavily damped. Depending on the amount of damping, typical ring times may last from 10 to 1000 cycles in length [14].

Although using only one transducer reduces a design's parts count, a disadvantage of the system is that one-transducer operation, combined with transducer ringing effects, limits the minimum

14

detectable target range. Since the receiver must be disabled during transmission and for a fixed amount of time immediately after transmission, the minimum range is given by

$$\text{minimum range (one transducer)} = V_{air}(\frac{T_{trans} + T_{ring}}{2}), \quad (2.6)$$

where V_{air} is as before, T_{trans} is the time duration of the transmitted pulse, and T_{ring} is the duration of the transducer ring.

The Polaroid Ultrasonic Rangefinder [15] is a good example of the one-transducer ultrasonic pulse-echo mode of operation. However, in this implementation the operating frequency is not fixed, but is a "chirp" of frequencies that include $f = 49.7, 53, 57,$ and 60 kHz. This selection of frequencies was chosen so that a loss in the reception of any one frequency, due to the acoustic properties of the reflecting surface, would not impair the overall ranging ability of the device.

The minimum range difficulty of one-transducer operation can be overcome by using separate transducers for the transmitter and receiver. However, the receiving transducer must now be acoustically shielded from the transmitting transducer or the receiver must be disabled during transmission. This prevents the transmitted ultrasound from directly coupling into the receiver and causing a false detection.

An example of the two-transducer pulse-echo mode of operation can be found in [16]. In this example an array of ultrasonic sensors are used to guide a wheeled robot along a winding hallway.

15

2.6.2 Phase-Delay Measurement

The phase-delay measurement technique measures target range with ultrasound by transmitting a continuous fixed-frequency wave and then measuring the resulting phase shift of the received reflected wave. Thus, separate transmitting and receiving transducers are required. The phase shift may be measured with a zero-crossing detector and a timer. The timer is reset when the transmitted (or reference) wave passes through a zero reference point as determined by the detector. The timer is then stopped when the received wave passes through the same zero reference point. The resulting elapsed time between the two zero crossings thus determines the target range exactly as in the pulse-echo technique described above.

A major disadvantage of the phase-delay measurement technique is that an ambiguity exists when the elapsed time becomes larger than the period of the transmitted frequency. This ambiguity thus limits the maximum range of this technique to

$$\text{maximum range} = V_{air} \left(\frac{1}{2 \, F_0} \right) \quad , \qquad (2.7)$$

where F_0 is the operating frequency of the transmitter.

2.6.3 Frequency-Shift Measurement

A third ultrasonic technique that has been used in blind guidance aids [3] is implemented as a continuous-wave frequency-modulated system. In this system the transmitted wave has an instantaneous frequency which depends only upon a sweep generator whose output is a

16

linear function of time. Target range is calculated by determining the frequency difference of the received wave and the wave currently being transmitted. This difference is easily generated using a multiplier and low-pass filter.

Since the frequency sweep is linear in time, the frequency difference indicates a time difference which represents the round-trip time of the transmitted-reflected signal. The round-trip time is

$$\text{round-trip time} = \frac{D_f \, T}{F_u - F_\ell} \tag{2.8}$$

where D_f is the frequency difference, T is the period of the frequency sweep, and F_u and F_ℓ are the upper and lower frequency limits of the transmitter, respectively. The range is then given by

$$\text{range} = V_{air} \left(\frac{\text{round-trip time}}{2} \right) \, . \tag{2.9}$$

This system has a maximum range given by

$$\text{maximum range} = V_{air} \left(\frac{T}{2} \right) \, , \tag{2.10}$$

since a range ambiguity will exist if the elapsed time exceeds the sweep period.

Instead of using pressure waves (ultrasound) to measure target range, alternate methods of measurement utilize electromagnetic radiation. Two portions of the electromagnetic spectrum are of interest; namely, infrared radiation and microwave radiation. As in

17

section 2.5, there are several optical implementations available that use infrared light; these are discussed in section 2.7. In section 2.8 the application of microwave radiation to target ranging will be discussed.

2.7 Optical Ranging

As with optical sensing methods that use a binary mode of detection, optical ranging methods for larger target ranges, i.e., greater than one centimeter, usually employ infrared light sources and detectors. Optical ranging can be implemented with several techniques, including triangulation, reflectance measurement, and phase-delay measurement.

2.7.1 Triangulation

Triangulation is a technique in which a light source/receiver assembly is mounted in such a way as to illuminate the target in question. Usually the receiver is a multiple array of phototransistors configured so that sensitive volumes are generated which lie at the intersection of the source and receiver's optical axes. See Figure 2.3 for a typical source/receiver arrangement. When a plane target surface lies normal to the source's optical axis and within a sensitive volume, the light is reflected uniformly from the surface in a semi-spherical pattern. Since each receiver has its own focusing lens, only a portion of this reflected light will be detected by a receiving element [17].

Note that in Figure 2.3 a different element in the receiver is activated depending upon the target position. Since the geometry of

18

Figure 2.3

Typical Geometry Used in Optical Triangulation Sensing Technique.

the situation is known (d, s, α, and β), the X position of the sensitive volumes V_1, V_2, ... , V_n are also known. Thus, a linear target displacement in the X direction can be determined using the array output to index a look-up table or its equivalent, such as a hardwired decoder.

The linear resolution of the sensor can be increased by increasing the number of light-sensing elements per unit area. A digital charge-injection device (C.I.D.) camera is a good example of a high resolution (244 x 248 pixels) two-dimensional optical receiver array. Using triangulation techniques, such cameras are used in [18] to locate the three-dimensional location of a projected laser spot.

2.7.2 Reflectance Measurement

Reflectance measurement is an analog ranging method which senses target range by measuring the output voltage of a receiving photodetector. For a given target reflectance and orientation, the received light flux is a function of distance [19]. A typical voltage response curve is shown in Figure 2.4. Note that target surface properties (such as reflectance) determine which curve is generated.

If the nature of the target surface is known a priori, the curves shown in Figure 2.4 may be used to generate a reference table. Measured receiver voltages then index the table to determine a corresponding target distance. To eliminate ambiguities, only one-half of the bell-shaped curve is used.

The reflectance-measurement technique has a severe disadvantage in that the amplitude of the output signal is solely used to determine the target distance or range. This is in contrast to the triangulation

20

Photodetector Output
(volts)

plane surface

plane surface with different reflective properties

Separation Distance

Sensing Range

21
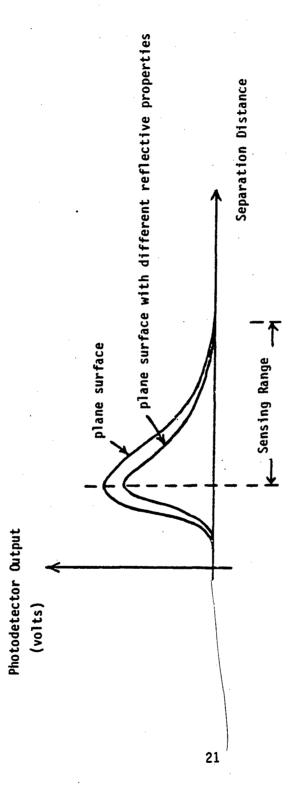
Figure 2.4

Photodetector Response as a Function of Distance and
Target Surface Properties.

scheme in which a receiving element was either "on" or "off" and the resulting binary pattern of several outputs determined the range. However, this difficulty has been reduced in [19] with digital signal processing techniques in which a weighted iterative filter is used to generate usable values of target range.

2.7.3 Phase-Delay Measurement

The optical phase-delay technique of target range measurement is identical in theory to the ultrasonic phase-delay measurement technique in that the phase shift of the reflected signal is measured and compared with the transmitted (reference) signal. However, a measurement problem exists in that the frequency of infrared light is approximately 100 THz (100 x 10^{12} Hz). At this frequency it is extremely difficult to determine a phase shift while at the same time providing an acceptable value of range resolution. This difficulty can be resolved by amplitude-modulating the light beam with a much lower frequency [20]. Once the high-frequency carrier is stripped away, phase detection on the remaining envelope may be performed.

2.8 Microwave Ranging

Microwave ranging, commonly known as radar (RAdio Detecting And Ranging), can be implemented in several forms as was ultrasonic ranging in section 2.6. When target distances are large, e.g., miles, the pulse-echo technique can be used since the delay time for such distances is measurable. For smaller distances an amplitude-modulated continuous-wave (AM-CW) system may be used, as in section 2.7.3 (phase detection). A variation of this method is the so-called

22

"two-frequency" radar in which three frequencies are used to measure target range:

$$F_1 = F_c - \Delta f, \tag{2.11}$$

$$F_2 = F_c \quad , \text{ and} \tag{2.12}$$

$$F_3 = F_c + \Delta f. \tag{2.13}$$

In this notation F_c and Δf represent the carrier and difference frequencies, respectively. This ranging method has been used to determine the lateral position of an automatically controlled automobile [21]. In this example F_c and Δf were chosen to be 10.5 GHz and 300 MHz, respectively. The transmitted signal consists of the two frequencies F_2 and F_3. By a clever modulation scheme in which both the transmitted and received signals are modulated by F_1 and F_2, after filtering, a pair of signals are generated whose phase difference is a function of target range only. In this example the values of F_c and Δf limit this range to approximately 50 cm.

Chapter 3

DEVELOPMENT OF A 40 KHZ ULTRASONIC RANGEFINDING SYSTEM

3.1 Selecting a Sensor System

The selection of any sensor system is based upon several
determining factors:

       1) method of sensing (e.g., pneumatic, inductive, etc.)

       2) effective sensing range

       3) sensor size and required mounting configuration

       4) support equipment required

       5) sensor reliability and durability

       6) system noise immunity

       7) system cost, etc.

The selection of a system based upon one or more of the above criteria
will usually limit the choices available in the evaluation of the
remaining criteria; e.g., if a pneumatic sensing system is chosen then
the effective sensing range will be limited to a few centimeters.

In the application discussed in this thesis, the modification of
Hexapod foot trajectories based upon terrain information requires the
anticipation of ground contact. Hence, a non-contacting sensor
system is clearly called for. Since a leg may travel a maximum

24

vertical distance of approximately two feet, the sensor system chosen must be able to gage foot altitudes over this range. As the space available on the Hexapod is quite limited, the sensor system must also be physically small and configurable in such a way that the sensors do not restrict the Hexapod's movement. The Hexapod carries its own regulated voltage supplies ($\pm$ 15 volts and +5 volts); ideally the sensor system selected should be able to work from these levels without the necessity of other supplies. The system should also be rugged and be able to withstand a reasonable amount of punishment without serious performance degradation. Furthermore, the sensing technique chosen should ideally be insensitive to external terrain conditions, and finally, the system should be as inexpensive as possible without sacrificing system performance.

The Hexapod's large effective sensing range requirement thus eliminates the pneumatic, capacitive, and inductive sensing techniques discussed in Chapter 2, which have maximum sensing ranges in the centimeter range. Microwave radar, also discussed in Chapter 2, exhibits problems both with excessively large minimum range and with the physical size of the sensor system itself, which in the case of Bishel [21] was quite large. There are two remaining methods from Chapter 2 that can meet the first three requirements listed above. These are the ultrasonic and the optical ranging methods. Due to the simplicity of the ultrasonic pulse-echo technique, several sensor system designs using this technique were investigated, two of which make use of the National Semiconductor LM1812 Ultrasonic Transceiver. The third design is a custom design built with operational amplifiers and discrete components. It is this design which has been implemented on the OSU Hexapod.

25
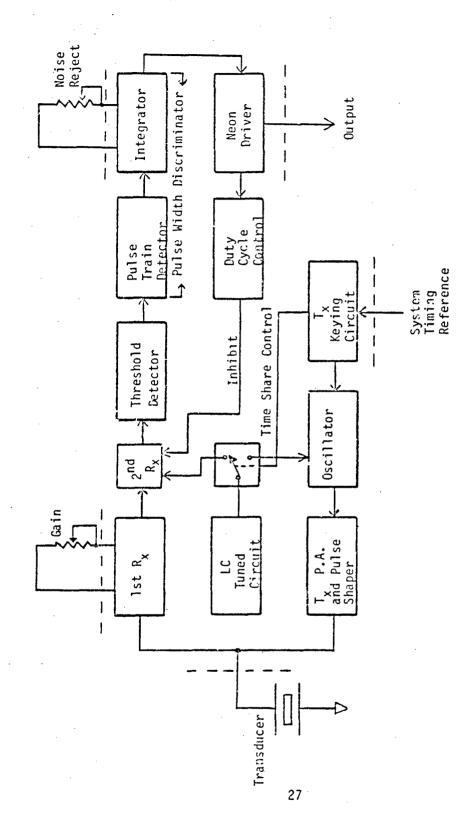
3.2 The National Semiconductor LM1812 Ultrasonic Transceiver

The LM1812 Ultrasonic Transceiver is a linear integrated circuit developed by the National Semiconductor Corporation in the mid 1970's [22]. Primarily intended for high frequency (200 kHz) use in fish-finding applications, it may also be used in air at the same or a lower frequency, such as 40 kHz. (In air, a lower frequency minimizes the effects of high frequency attenuation.) Features of this circuit include single-transducer operation, high power output (12 watts maximum), and internal noise rejection. A block diagram of the transceiver is shown in Figure 3.1. National Semiconductor-suggested circuits for the 200 kHz (water) and 40 kHz (air) implementations are shown in Figures 3.2 and 3.3, respectively [23]. Note that in both cases the transmitted frequency is a function of only one tuned circuit, L1 and C1. A modification of the 200 kHz circuit of Figure 3.2 for use in air is shown in Figure 3.4 [14].

As mentioned in section 2.6.1, target range is calculated with the pulse-echo technique by determining the elapsed time from the beginning of transmission to the first received echo. This elapsed time is easily determined with an external counter which is reset upon transmission and then disabled upon the first reception of an echo. Pin 14 of the LM1812 transceiver is provided for just such a purpose: upon transmission pin 14 goes low and stays low until an echo is received, at which time pin 14 rises. Thus, with appropriate gating, the rising and falling edges of pin 14 may be used to drive the clear and enable lines of a counter.

3.2.1 LM1812 40 kHz Operation

A modification of the 200 kHz circuit of Figure 3.2 for use in air

26

Figure 3.1

Functional Diagram of LM1812 Transceiver.
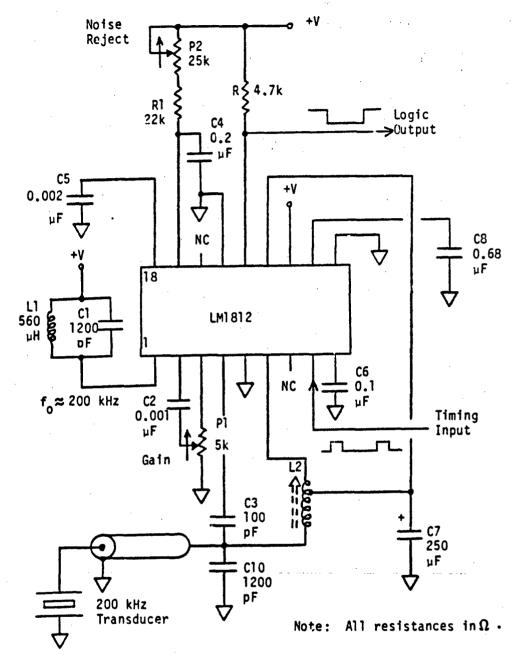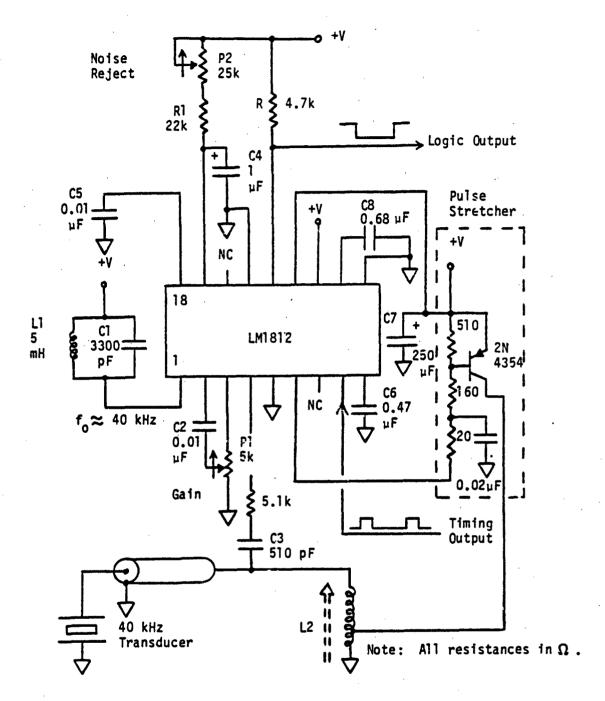
Figure 3.2

LM1812 200 kHz Underwater Application.

28

Figure 3.3

LM1812 40 kHz Operation in Air.

29
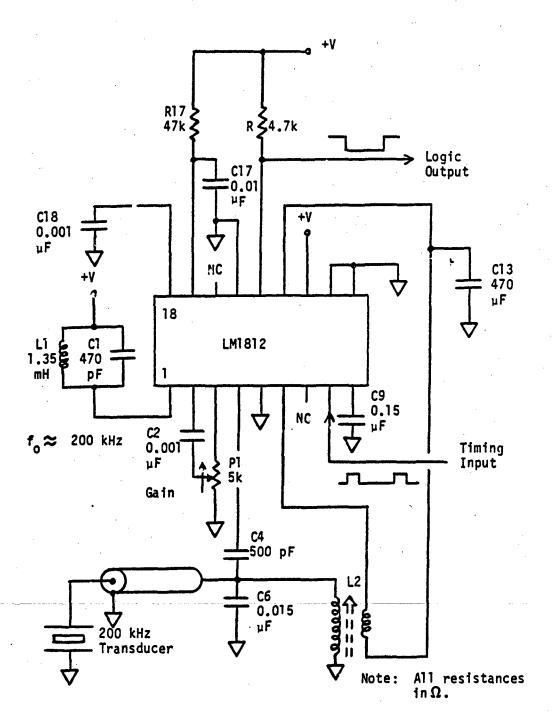
Figure 3.4

LM1812 200 kHz Operation in Air.

at 40 kHz is shown in Figure 3.3. The ultrasonic transducer used in this circuit is a piezo-electric transducer having a resonant frequency of 40 kHz. Available from Massa Products Corporation, its complete specifications are listed in Figure A.1.

The circuit of Figure 3.3 was breadboarded. During testing, it was noted that following transmission, the transducer exhibited a mechanical ringing whose envelope closely approximated a decaying exponential. This ringing was found to falsely trigger the receiver circuitry. Since the receiver in the LM1812 circuit is internally enabled as soon as the transmitter is disabled, the small amount of transducer ring is electrically coupled to the receiver input. This electrically-coupled mechanical ringing consistently generated a false detection signal at the transceiver output, pin 14. An attempt to reduce the duration of this transducer ringing met with only limited success, the basic idea being to briefly connect an electronic damping circuit across the transducer immediately following transmission. An alternative to electronic damping is to increase the internal mechanical damping of the transducer, i.e., use a different transducer. However, other 40 kHz transducers were difficult to locate; hence, this circuit was abandoned with hopes that operation at 200 kHz would give better results.

3.2.2 LM1812 200 kHz Operation

Although the LM1812 transceiver was designed to operate at a frequency of 200 kHz in water, it may be operated at this frequency in air (see Figure 3.4) provided that the reduction in maximum range is acceptable. This reduction is due to the attenuation in air of the 200 kHz pressure waves. Operation at this frequency requires a transducer

31

having a 200 kHz resonant frequency. This transducer is also available from Massa Products Corporation; its complete specifications are given in Figure A.2.

The breadboarded circuit of Figure 3.4 gave good results; the duration of transducer ring was short enough that the false detection plaguing the 40 kHz circuit did not occur. With component values shown this circuit is capable of detecting objects up to approximately ten feet away, providing the reflecting surface is flat and is positioned parallel to the transducer face.

Due to the very narrow beamwidth of the 200 kHz transducer, slight ($\pm 5°$) angular deviations from the beam axis resulted in a total loss of reflected signal. Similar results were obtained by Canali and De Cicco [24] in a temperature-compensated 250 kHz version of this same circuit. Since any leg-mounted sensor on the OSU Hexapod will be subjected to angular displacements in the range of $\pm 20°$, the 200 kHz system as discussed above was deemed unusable in this application. Inspection of the radiation pattern of Figure A.1 provides a clue as to a possible solution to this problem. At 40 kHz the transducer beamwidth is relatively wide ($\sim \pm 10°$ at the -3 dB points), hence, the wider transducer beamwidth should permit a larger angular deviation from the beam axis without a total loss of reflected signal. A discussion of the received signal strength as a function of beamwidth, angular sensor displacement, etc. is given in section 4.2. To test this theory a 40 kHz ultrasonic transmitter/receiver circuit was constructed with operational amplifiers and discrete components. This circuit is described in detail in the following section.

3.3 A 40 kHz Ultrasonic Transmitter/Receiver Design

The transducer ringing problem experienced with the 40 kHz LM1812 circuit (section 3.2.1) can be circumvented using separate transmit and receive transducers and custom electronics. Recall the problem with the single transducer circuit: following transmission, the mechanical ring was electrically coupled into the receiver. In this design, since separate transducers are used, the electrical coupling between the transmitting transducer and the receiver input is eliminated. Thus, provided that the transducers are acoustically isolated, the receiving transducer may be enabled while the transmitting transducer is still ringing. Acoustic isolation is required so that the transmitter ring will not be acoustically coupled into the receiving transducer. If this were to occur then the receiving transducer would treat the coupled pressure waves as ordinary echos, hence, a false detection signal would be generated.

The block diagrams of the transmitter and receiver circuits are given in Figures 3.5 and 3.6, respectively. In the transmitter a free-running 40 kHz oscillator is gated through an analog switch. The control (or gating) signal is the output of a pulse width generator (monostable multivibrator) which is, in turn, enabled by the START signal from the digital circuitry. The resulting output of the analog switch is a one millisecond "burst" of 40 kHz ultrasound (40 cycles of 40 kHz). After power boosting, this burst is sent to the transmitting transducer via shielded cable.

The receiver is similarly simple; the output of the receiving transducer is preamplified and then bandpass filtered. The bandpassed
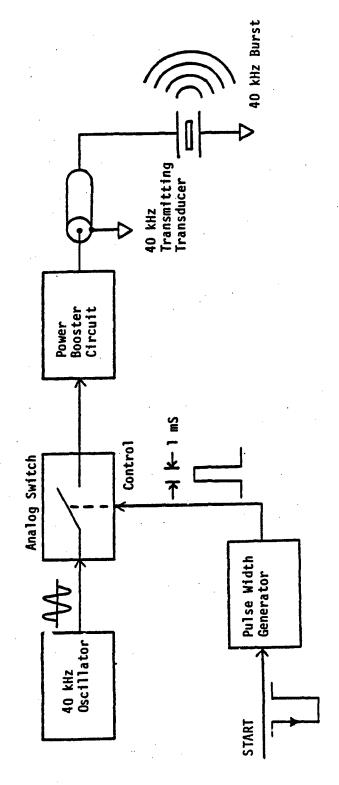
Figure 3.5

Block Diagram of 40 kHz Custom Transmitter.

34

Figure 3.6

Block Diagram of 40 kHz Custom Receiver.

signal is then incoherently demodu... ted and the resulting envelope is threshold detected to form a STOP signal. After detection, this STOP signal is converted to TTL- (transistor-transistor-logic) compatible voltage levels and is then used to disable the counter in the digital circuit. This is exactly the same procedure followed in the LM1812 transceiver circuit discussed previously. For clarity, the interaction between the analog and digital circuitry is shown in Figure 3.7. Although this figure shows only one transmitter/receiver pair, hereafter referred to as a "sensor", it is to be noted that in this application the Hexapod is equipped with six such sensors, one per leg.

3.3.1 Transmitter Circuitry

A detailed schematic of the transmitter circuit used in the realization of Figure 3.5 is shown in Figure 3.8. Given the gain-bandwidth requirements of the transmitter and receiver (Figure 3.9) subcircuits, National Semiconductor LF351 operational amplifiers were chosen.

Two such amplifiers are used in the oscillator circuit. Oscillator component values were chosen so as to achieve a nominal operating frequency of 40 kHz, which may be fine-tuned with potentiometer P1. The amplitude of the oscillator output may be adjusted with potentiometer P2.

The gating of the oscillator is controlled by a monostable multivibrator (1/2 74LS123). Upon reception of a negative-going START edge at pins 3 and 4 from the digital circuit, a pulse is generated whose t...e duration is nominally fixed at one millisecond. This width may be adjusted with potentiometer P3. The outputs of the monostable

36
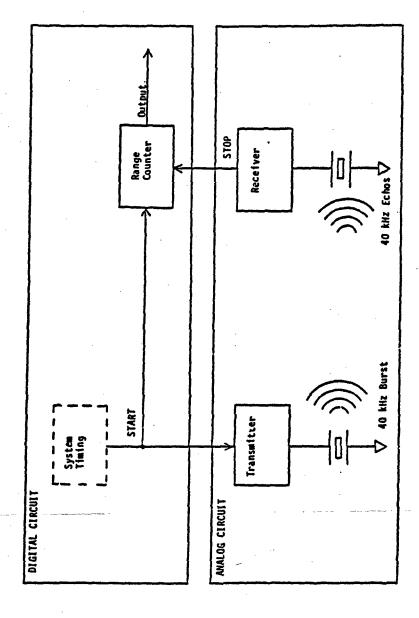
Figure 3.7
Analog/Digital Circuit Interaction.

37
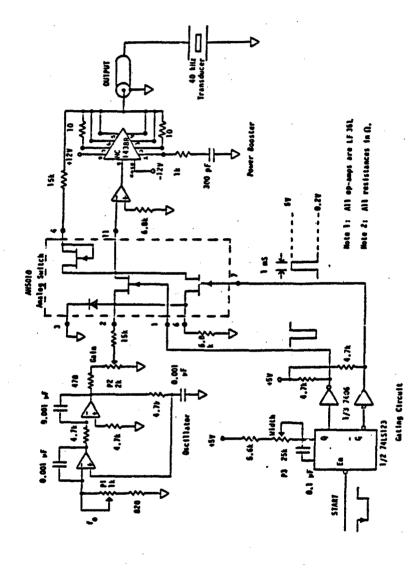
Figure 3.8

40 kHz Custom Transmitter Circuitry.

38
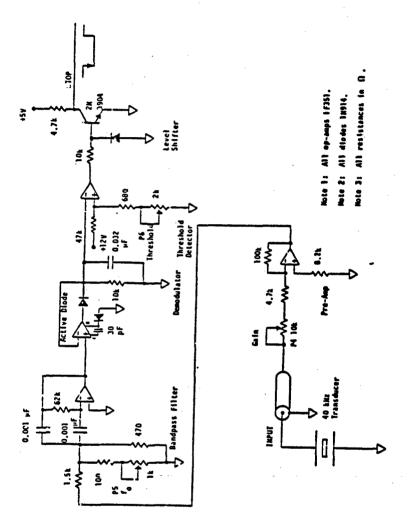
Figure 3.9

40 kHz Custom Receiver Circuitry.

39

multivibrator drive TTL open-collector inverters (1/3 7406) which in turn drive the control lines (pins 1 and 7) of the analog switch. The analog switch (National Semiconductor AH5010) is internally configured as a four-input, one-output multiplexer; however, in this circuit only two inputs are used. Thus, its output (pin 11) will consist of either the oscillator input (pin 2) or a ground input (pin 6). The two switch control lines are active low, and as wired in this circuit are complementary. In this way when the oscillator is disconnected the power booster input will be effectively grounded. Open-collector logic is used to provide a large value of turn-off voltage required by the switch control inputs.

The output of the analog switch drives a power booster circuit consisting of an LF351 op-amp cascaded with a Motorola MC1438R current amplifier. A 15 kΩ feedback resistor is connected from the booster output to pin 4 of the analog switch. The internal FET transistor between switch pins 4 and 11 compensates for the temperature sensitivity and "ON" resistance of the FET switches. Since switch input pin 2 is connected to the oscillator circuit through a 15 kΩ resistor, the overall voltage gain of the power booster circuit (from pot wiper to booster output) is unity. In the event of a short circuit at the booster output, two 10 ohm, 2 watt resistors (connected across pins 2 and 7 and the output, respectively) limit this short-circuit current to approximately 300 mA.

3.3.2 Receiver Circuitry

The realization of the receiver block diagram (Figure 3 6) is shown in Figure 3.9. The 40 kHz echo is first preamplified with an

inverting amplifier whose gain may be adjusted with potentiometer P4. The component values chosen fix the maximum and minimum values of preamp gain to approximately 21 and 7, respectively.

The output of the preamp drives the active bandpass filter. Its schematic and voltage transfer characteristics are given in Figure A.3. Component values were chosen for a center frequency, F_0, of 40 kHz and a midband voltage gain, H_0, of 20. The bandwidth, BW, was chosen to be approximately 5 kHz, giving the filter a Q value of eight. (For this filter configuration the upper limit on Q is approximately ten.) Thus, from Figure A.3,

$$R_1 = 1.59 \text{ k}\Omega, \tag{3.1}$$

$$R_2 = 294 \ \Omega, \text{ and} \tag{3.2}$$

$$R_3 = 63.66 \text{ k}\Omega, \tag{3.3}$$

if $C_1 = C_2 = 0.001 \ \mu F$. Note that if the bandwidth is fixed then resistor R_2 alone determines the filter's center frequency. Hence, in Figure 3.9, the 470 ohm shunt across the series combination of potentiometer P5 and the 100 ohm resistor permits the filter's center frequency to be fine-tuned, avoiding the necessity of close-tolerance filter components.

The output of the filter is a relatively large amplitude burst of 40 kHz ultrasound which swings symmetrically about ground. The leading edge of this burst denotes the first reception of the received echo; it is this edge which must be detected and converted into a digital STOP pulse for use by the digital circuitry. A simple incoherent demodulator/threshold circuit is used to perform this conversion. After

41

demodulation, only the envelope of the positive portion of the burst remains, which is then passed through a threshold detector. The detector discriminates between valid returned echos and spurious noise which may be present at the demodulator output.

A simple diode demodulator consists of a diode followed by a low-pass filter. A conventional silicon diode must be forward biased by at least its cut-in voltage (approximately 0.6 volts) before conduction can occur; hence, for the demodulator output to track a rising input waveform the demodulator input must be larger than the demodulator output by at least the diode cut-in voltage. In any ultrasonic ranging system the received signal strength decreases greatly with increased target range, and may easily drop to a value below that of the diode cut-in voltage. This occurrence severely limits the system's maximum sensing range.

A simple solution to the above difficulty is to use an active diode. Ideally, the demodulator input then need only be greater than its output for signal rectification to occur. The active diode configuration thus permits rectification even with very small (millivolt) comparator voltage differences.

The required time constant of the demodulator's low-pass filter is determined by the operating frequency of the transmitter. This time constant is chosen so that the filter can follow the low-frequency envelope of the received echo while rejecting the high-frequency (40 kHz) carrier. A time constant of 320 μS was experimentally found to optimally satisfy both requirements.

The output of the demodulator circuit is compared to a DC voltage value in the threshold circuit. With component values shown the

reference level may be varied with P6 from approximately 0.17 to 0.65 volts, a range found experimentally to be adequate. The output of the threshold circuit is a pulse swinging from -12 to +12 volts as the demodulator output rises above the DC threshold value. This pulse is then converted to TTL-compatible voltage levels using a simple transistor inverter with its collector pulled up to +5 volts. A diode clamp across the transistor's base-emitter junction prevents reverse breakdown during large negative swings in the output of the threshold circuit.

3.4 Digital Electronics Design of the 40 kHz Ultrasonic Rangefinder

The digital electronics of the ultrasonic rangefinder is comprised of three basic subsystems:

1) System timing and transmitter control
2) Range counter and counter control
3) PDP 11/70 computer interface logic.

3.4.1 System Timing and Transmitter Control
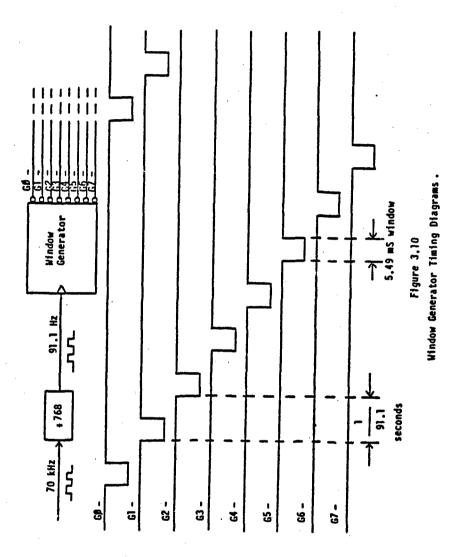
The system timing and transmitter control subsystem specifies which ultrasonic sensor is to be enabled and when this enabling is to take place. A sequencing scheme is followed in which only one transmitter/receiver pair is active at any given time. This sequencing is used to minimize any acoustic interaction between sensors, and is implemented with a "window generator" circuit. A low frequency clock signal derived from the range counter clock signal (see section 3.4.2) drives the window generator such that its outputs G0-G7 (see Figures

43
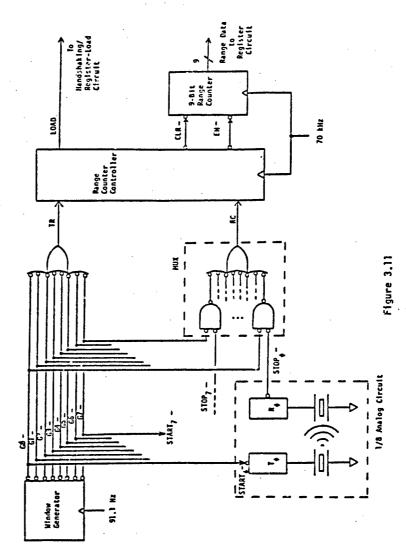
3.10 and 3.11) sequentially enable one of eight possible transmitters approximately 11.4 times per second each. Again, in this application, only six sensors are, in fact, being used. Note from Figure 3.11 that a falling edge on window generator output line Gn is required to trigger the $START_n$ input of ultrasonic transmitter T_n. Range data is computed each time a sensor is enabled. Since the Hexapod's maximum speed is approximately five inches per second, the 11.4 Hz range update rate was deemed satisfactory; larger Hexapod velocities would require faster scanning speeds.

3.4.2 Range Counter and Counter Control

Since the sensors are scanned sequentially, the elapsed-time counter, hereafter referred to as the range counter, may be shared by all sensors. Thus, as shown in Figure 3.7, the counter begins to count as each transmitter is enabled. A $STOP_n$ pulse from corresponding receiver R_n (see Figure 3.11) disables the counter and loads data register DRn with the counter value. If a STOP pulse is not received from the corresponding receiver within a time corresponding to a maximum target range of approximately 37 inches, then the counter is automatically disabled. This maximum range value is a function of the window generator clock frequency, and 's derived in the following paragraphs.

The range counter controller of Figure 3.11 has two inputs, transmit, TR, and receive, RC. TR is generated at a fixed rate of 91.1 Hz from the logically OR'ed outputs of the window generator circuit. RC is the output of an eight-input multiplexer which routes the STOP signals from the receiver circuits. The selection of the

44

Figure 3.10
Window Generator Timing Diagrams.

45

Figure 3.11   Simplified View of Transmitter and Range Counter Control Logic.

proper receiver output is determined by window generator outputs GO thru G7. The counter controller has three outputs, CLR, EN, and LOAD. When CLR is asserted the counter is synchronously cleared; when EN is asserted the counter is enabled and begins to count. The LOAD output synchronously loads one of eight data registers with the range counter value provided that the computer is not simultaneously trying to read that particular register. (Register loading is discussed fully in section 3.4.3.)

The state diagram of the range counter controller is shown in Figure 3.12. State A is the counter reset state; while in this state the counter is continually cleared. Controller input TR becomes ̄.̄ically true at the time that any window generator output falls, i.e., at the time that one of the transmitters is enabled. At this time the counter controller synchronously switches from state A to state B. While in state B, the count state, the counter controller asserts its EN output and the previously cleared range counter begins to count. This count function is disabled if the counter controller's RC input is asserted (a STOP signal has been received) or if the TR input is unasserted (target is out of range). From Figures 3.10 and 3.11, TR is unasserted after a period of 5.49 milliseconds. This time, established by the window generator circuit and its 91.1 Hz clock, determines the maximum sensing range of the system:

$$\text{Maximum Range} = \text{Maximum Delay Time} \times V_{air}/2 \qquad (3.4)$$
$$= 5.49 \text{ mS} \times 1130.1 \text{ ft/sec} /2$$
$$= 3.1 \text{ ft} = 37.2 \text{ inches.} \quad (T = 22 \text{ °C.})$$

This 37.2 inch maximum range is more than sufficient for the OSU

47
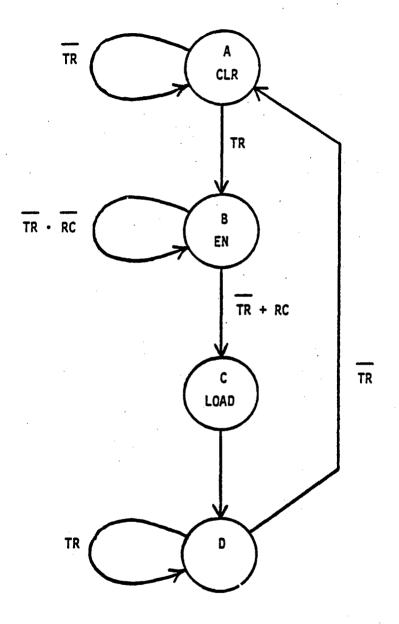
Figure 3.12

State Diagram of Range Counter Controller.

48

Hexapod application. Continuing with the state diagram of Figure 3.12, if either of the two branch conditions discussed above exist during state B, the counter controller synchronously switches to state C, the load state. During the clock cycle that the counter controller is in this state, the controller's LOAD output is asserted and the range counter data is loaded into one of eight data registers, as discussed in section 3.4.3. Following state C, the controller automatically enters state D, the wait state. The controller remains in this state until the TR input is unasserted, at which time the controller begins an entirely new cycle by transferring back to state A. If the out-of-range condition forces the transfer to state C, then the controller will remain in state D for only one clock cycle before transferring back to state A.

The range counter and counter controller clock rate was chosen so as to provide a "reasonable" amount of range resolution, which for the Hexapod was chosen to be 0.1 inches. Since the resolution is given by

$$\text{Resolution} = \frac{V_{air}}{2 \times \text{clock frequency}} \qquad (3.5)$$

this resolution corresponds to a clock frequency of 67.8 kHz. (T = 22° C.) For simplicity, a clock frequency of 70 kHz was chosen. This clock signal is derived from the high frequency clock used in the computer interface circuitry of section 3.4.3. Since from Figure 3.10 the maximum delay time during which the counter is enabled is 5.49 milliseconds, a frequency of 70 kHz will produce a maximum counter value of 384. Hence, a 9-bit binary counter is required.

49

3.4.3 PDP 11/70 Computer Interface Logic

The computer interface logic is responsible for communication between the digital circuit and the PDP 11/70 minicomputer. Several communication schemes to retrieve the updated range information are possible, including device polling and interrupt-driven techniques. A third, much faster method which has no computer-interface handshaking relies upon the digital circuit to be ready for computer read and write operations at any given time. Since the cycle time through the existing Hexapod control loop is already quite long, it was desired that the I/O interaction time between the 11/70 and the digital circuit be minimized. Hence, the third communication technique, described below, was chosen.

Communication with the digital circuit is implemented with a 16-bit parallel data link which interfaces with the 11/70 through a DEC DR11-C parallel interface board. The DR11-C interface board handles all computer-requested input/output (I/O) with two signal lines labeled NEW DATA READY, NDR, and DATA TRANSMITTED, DT [25]. During an 11/70 write operation, 16-bit data is written to the DR11-C output port as the NDR line is strobed. The falling edge of NDR indicates that the data is stable. During an 11/70 read operation the 11/70 reads the 16-bit DR11-C input port and simultaneously strobes the DT line. The falling edge of DT indicates the time at which the data is latched. The asynchronous nature of this I/O operation is indicated in Figure 3.13.

Since the digital circuit has eight data registers, each nine bits wide, for a total of 72 data bits, a multiplexed I/O scheme is used. In this scheme the 11/70 first transmits a 3-bit register address

50

New Data Ready

H

L

Computer Output Data
is Stable

t

Data Transmitted

Input Data Latched
by Computer

t
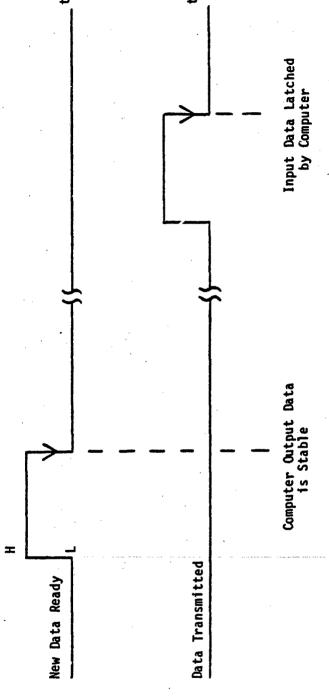
Figure 3.13

DR11-C I/O Signals.

51

which selects the desired data register, and then strobes NDR. The 11/70 then reads the selected 9-bit data and strobes DT. Note that since no handshaking is used a problem can exist when the computer attempts to read the contents of a data register which is in the process of being loaded. Since the data register flip-flops may still be changing state, the read operation could return unreliable data. A simple solution to this problem is to inhibit the register-loading operation during the time between the negative edges of the NDR and DT signals of Figure 3.13. The capability to perform and detect such a load-inhibit operation has been implemented in the digital circuit and is shown in block diagram form in Figure 3.14.

The basic premise of the register-loading scheme of Figure 3.14 is that data register DRn may only be loaded if three conditions exist:

1) the sequencing logic determines that register DRn is to be loaded,

2) the counter controller's LOAD output is asserted, and

3) the 11/70 is NOT trying to read register DRn at the time that (1) and (2) are occurring.

If all three conditions are satisfied then the LOAD DRn output (see Figure 3.14) is asserted. This output directly enables the loading of register DRn.

Condition (1) is satisfied by the "window expander" circuit shown in Figure 3.14. This circuit is driven by the window generator circuit; its input and output timing waveforms (G0-G7 and E0-E7, respectively) are shown in Figure 3.15. Thus, only one of the eight registers may be loaded at any given time. The outputs from this circuit are logically AND'ed with condition (2) (LOAD) and then with

52
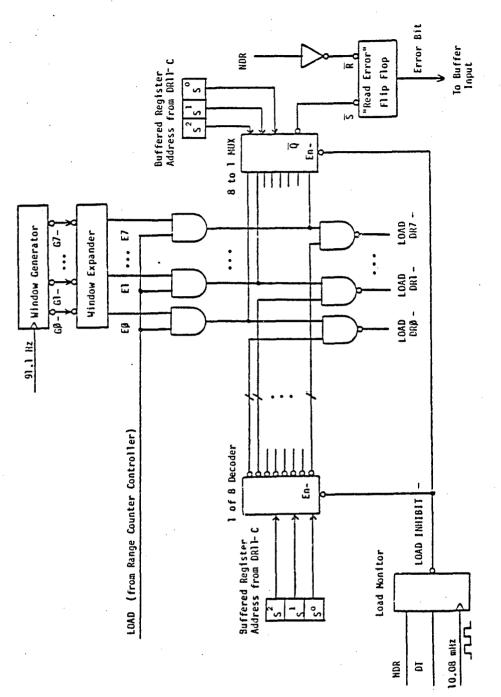
Figure 3.14   Handshaking/Register Loading Circuit.
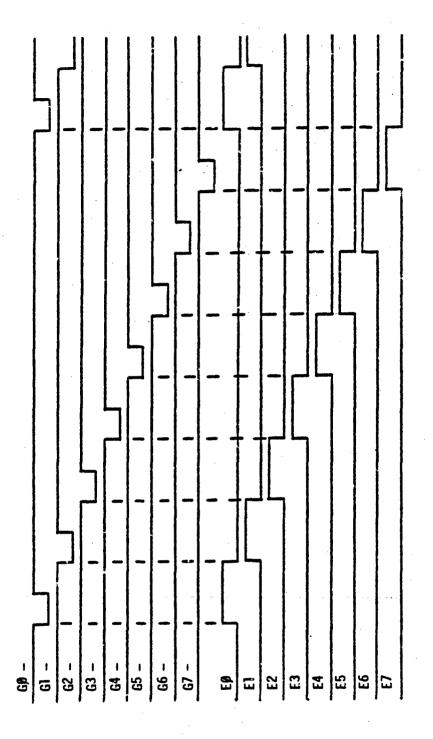
Figure 3.15

Window Expander Input and Output Timing Waveforms.

54

the outputs of a one-of-eight decoder which is enabled by the LOAD
INHIBIT output of the Load Monitor. Note that the decoder outputs are
high when the decoder is disabled, i.e., when the computer is not
performing a read or write operation. When the decoder is enabled and
register DRn is selected (by the computer), decoder output n falls,
disabling the LOAD DRn line.

The Load Monitor shown in Figure 3.14 is a simple controller which
monitors the NDR and DT lines for their respective negative edges
discussed above. Its state diagram is shown in Figure 3.16. Its output
is asserted between the times that the NDR and DT lines go negative.
Since the duration of the NDR and DT pulses of Figure 3.13 is very
short (typically 400 nS), the controller must respond very quickly in
order to avoid missing one or both pulses. This capability is provided
using a relatively high frequency Load Monitor clock. A conveniently
available 10.08 MHz hybrid integrated circuit clock chip provides this
frequency.

The detection of the load-inhibit operation is performed by the
eight-to-one multiplexer shown in Figure 3.14. When the multiplexer is
enabled each input line monitors the logical AND of conditions (1) and
(2). From above, if the computer selects register DRn, then multiplexer
input line n is routed to the output which will set the "read error"
flip-flop if input line n is a logical "1". Note that this flip-flop is
reset upon every rising edge of NDR, i.e., during a new read cycle.

The remainder of the interface logic consists only of data
registers DR0-DR7, multiplexers M0-M8, and the isolation circuitry
which buffers all I/O lines to and from the DR11-C interface board.
Multiplexers M0-M8 channel the nine bits of computer-selected data to

55
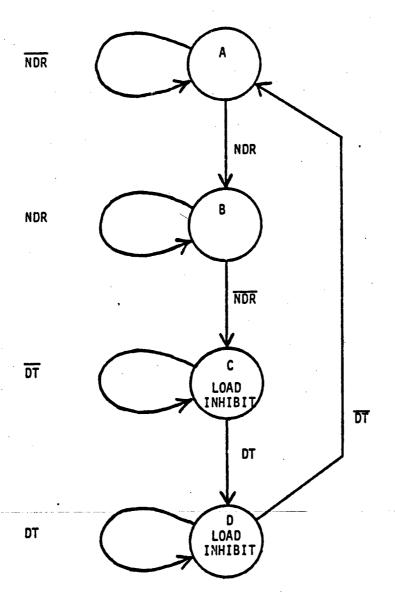
Figure 3.16
Load Monitor State Diagram.

56

the input port of the DR11-C. This logic is shown in Figure 3.17. The buffer circuitry consists of Motorola MC1488 line drivers and Hewlett-Packard HP2630 opto-isolators. This circuitry is shown in Figure 3.18.

A detailed schematic of the digital circuit as discussed above may be found in Appendix C.
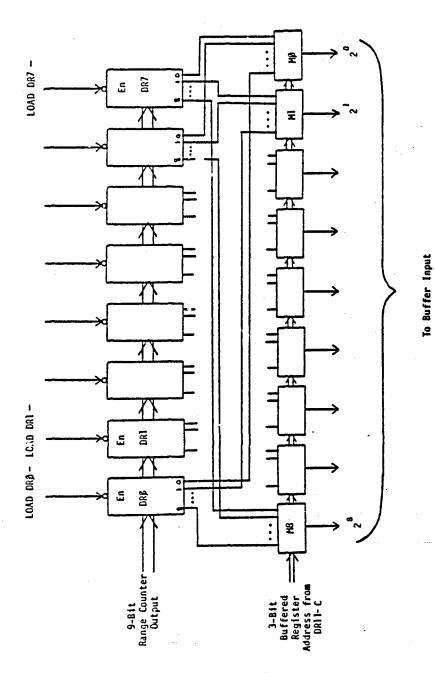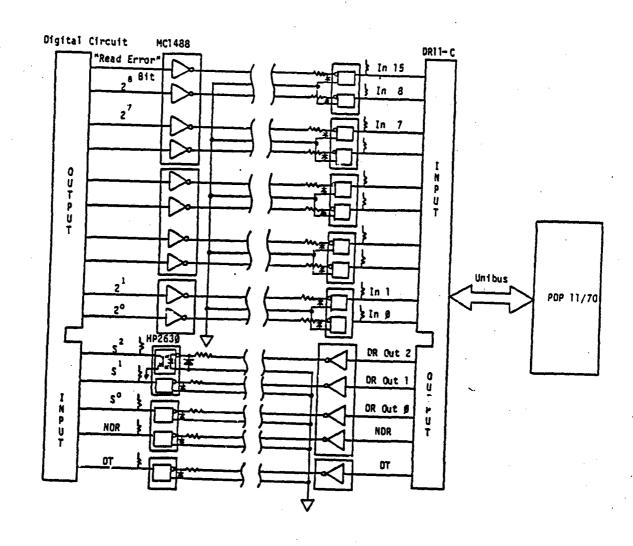
Figure 3.17

Data Register and Multiplexer Circuit.

Figure 3.18

I/O Buffer Circuitry.

Chapter 4

EXPERIMENTAL RESULTS OF THE 40 KHZ ULTRASONIC RANGEFINDER

In this chapter, experimental observations of the 40 kHz ultrasonic rangefinding system developed in Chapter 3 are discussed. These observations can be separated into three groups, one dealing with the rangefinder's ability to measure target distance, another which discusses the rangefinder's environmental sensitivity, and the third which discusses the operation of the rangefinder's analog circuitry.

A photograph of the breadboarded rangefinding system is shown in Figure 4.1; note that six transmitters and receivers have been constructed. To reduce the amount of transmitter circuitry, a single 40 kHz oscillator is shared by each of the six transmitters. A prototype ultrasonic sensor unit, containing the transmit and receive transducers, is shown in Figure 4.2. The sensor's "nose" piece, which separates the transmit and receive transducers, is used to eliminate transducer sidelobe interference which would otherwise falsely trigger the receiver circuitry.

4.1 Rangefinder Measurement of Target Distance

The ability to gage target distance with the rangefinder developed
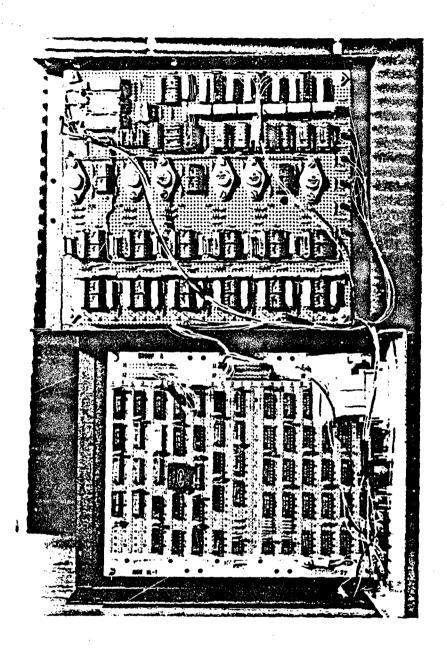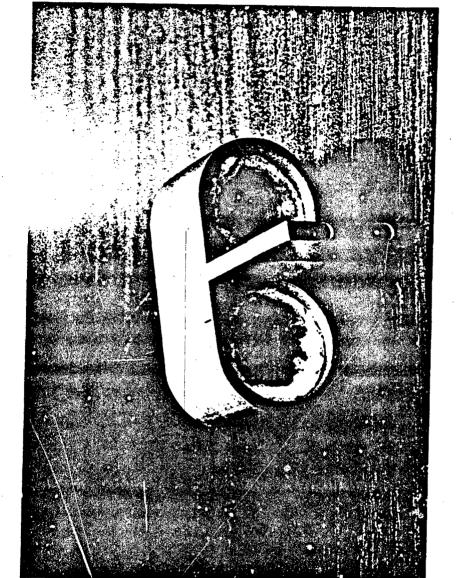
Figure 4.1
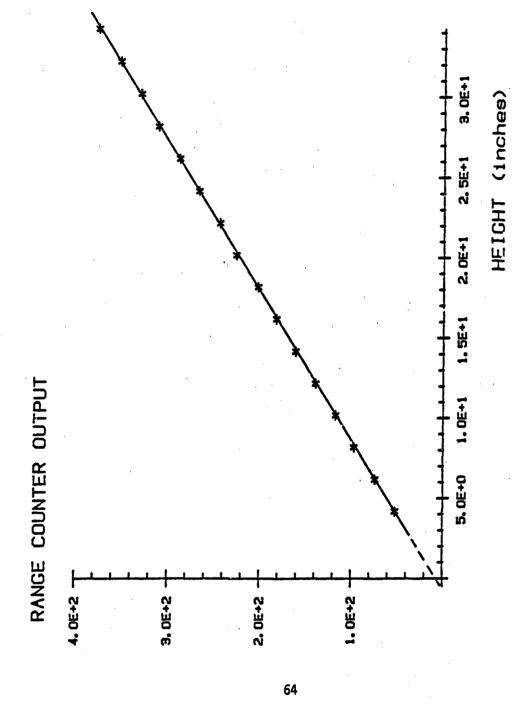
Analog and Digital Circuitry of the 40 kHz Ultrasonic Rangefinder.

Figure 4.2

Prototype Ultrasonic Sensor Unit.

62

in Chapter 3 was experimentally determined using the digital circuit's range counter. In this experiment, the 9-bit range counter output was monitored as the sensor was moved vertically above a flat metal plate. Figure 4.3 shows the experimental plot of the counter output as a function of vertical sensor height. Note the extreme linearity of the plotted points; a linear regression analysis of this data yields a correlation coefficient of greater than 0.99. (A straight line has a correlation coefficient of unity.) Also note from Figure 4.3 that the data does not pass through the origin, but exhibits an offset. This offset, corresponding to a measured delay of approximately 120 μS, is due to the fixed response time of the receiver circuitry. Finally, note from Figure 4.3 that the x-axis variable, sensor height, is plotted to only 35 inches. At greater heights the received echos are too weak to trigger the receiver circuitry. This is due to the fixed settings of the receiver's gain and threshold potentiometers, which may be adjusted for a larger maximum range if desired. However, the present maximum range of approximately 35 inches is more than sufficient for the OSU Hexapod, and in any case is practically limited to 37.2 inches by the digital circuit's timing restrictions (see section 3.4.2). Furthermore, increasing the gain to permit larger values of target range may present a problem when small target ranges are encountered, due to the large amplification of residual noise present at the output of the demodulator circuit. This amplified noise, when larger in magnitude than the DC threshold level, will generate erroneous STOP signals at the receiver output. With current gain and threshold settings this phenomenon is observable when a flat metal target is positioned approximately four inches from the sensor. On the OSU Hexapod, this difficulty is avoided

63

RANGE COUNTER OUTPUT

HEIGHT (inches)

Figure 4.3

Range Counter Output as a Function of Sensor Height.

64

by mounting each sensor more than four inches from the foot tip of each leg, as will be discussed in section 5.1.

4.2 Environmental Sensitivity of the 40 kHz Ultrasonic Rangefinder

As mentioned briefly in section 3.2.2, the rangefinder is sensitive to sensor/target orientation angle, sensor height above a target, and the scattering properties of the target itself. In each case, the net result is a reduction in the strength of the received signal, which is discussed in the following sections. Rangefinder sensitivity to the above parameters will be demonstrated by recording the peak amplitude of the envelope of the received sinusoidal echos; this envelope is available at the receiver's demodulator output. This is a convenient point at which to observe trends in the strength of the received signal, for at this point, the signal has already been amplified and had its carrier signal removed.

For convenience, in the following experiments an X-Y plotter was used to obtain a permanent record of the peak-amplitude data as a function of horizontal sensor position. The dependent variable was chosen so that the sensor could laterally scan the target surface at a constant height. This configuration thus permits a measurement of received signal strength as a function of target scattering properties.

The plotter's x-axis input is controlled by a potentiometer which records the ultrasonic sensor's horizontal position over a target. The generation of the plotter's y-axis input signal is described below.

Due to the plotter's very low bandwidth, the receiver's demodulator output cannot directly be used to drive the plotter's y-axis input. This is due to the plotter's inability to accurately

65

respond to the relatively fast demodulated signal, the time duration of which is typically only one to two milliseconds. To overcome this bandwidth mismatch, a peak-detect/hold circuit is used to interface the demodulator output to the plotter's y-axis input. The peak detector tracks and retains the maximum value of the demodulated signal. To detect time-varying changes in the peak envelope amplitude, and hence, in the peak strength of the received signal, the peak detector is synchronously reset by the digital circuit at its range-update rate of 11.4 Hz. A sample-and-hold circuit samples the peak detector output; its purpose is to mask the "glitches" present in the detector output due to the 11.4 Hz reset operation. The output of the hold circuit is a slowly varying DC voltage representing the peak value of the received signal; it is this signal which directly drives the y-axis input of the X-Y plotter.
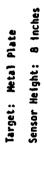
4.2.1 Sensor/Target Orientation Angle Sensitivity
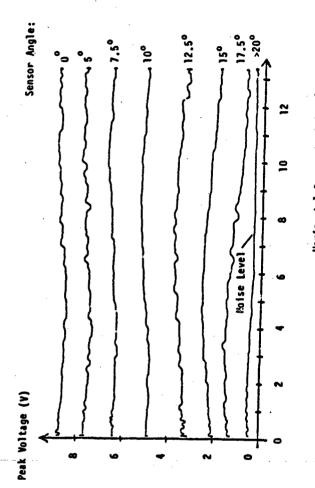
In this work, sensor/target orientation angle, hereafter referred to as sensor angle, is defined as the angle between a line drawn normal to the sensors' transducer faces and a line drawn normal to the target surface. See Figure 4.4. For simplicity, in the following experiments the sensor is limited to one degree of angular freedom. This sensor angle, from Figure 4.4, is generated by a rotation about point A.

Figure 4.5 illustrates the effect that sensor angle has upon the strength of the received signal. In this experiment, the target was a flat metal plate separated from the sensor by a distance of eight inches. To obtain an average indication of this peak signal strength, the sensor was swept horizontally at a slow rate above the metal plate.

66

Figure 4.4

Definition of Sensor Angle.

67

Figure 4.5

Peak Demodulator Output Voltage as a Function of Horizontal Sensor Position and Angle.

68

Note from Figure 4.5 the resulting decrease in peak signal strength as the sensor angle is increased. At an angle of ten degrees the received signal strength is approximately 60% of its value when the sensor angle is zero. At fifteen degrees the signal strength has been reduced to approximately 25% of its zero-angle value; at angles exceeding approximately twenty degrees, the signal is lost in the noise of the demodulator output.

The decrease in received signal strength as a result of increased sensor angle is not unexpected if the concepts of specular and diffuse reflection are considered. Specular reflection results when a wave, e.g., a sound wave, is incident upon a smooth surface. "Smooth" in this sense is defined by a surface whose irregularities are much smaller than the wavelength of the incident wave. On the other hand, diffuse reflection will occur if the dimensions of the surface irregularities approach or exceed the wavelength of the incident wave. As the wavelength of 40 kHz ultrasound is 0.34 inches, specular reflection is clearly occurring in the above experiment. However, this type of reflection is undesirable in a ranging system that will be subjected to large values of angular sensor displacement, a fact easily seen from Figure 4.6. Note that the receiver can only detect an echo if the reflected sound impinges upon the receiving transducer. Advancing this concept, if the transmit and receive transducers have narrow beamwidths, then only slight sensor angles can be tolerated. Conversely, wide transducer beamwidths will permit larger angular deviations before the echo is undetectable. This concept was instrumental in the selection of the wider-beamwidth 40 kHz transducers over the narrow-beamwidth 200 kHz units of section 3.2.2.

69
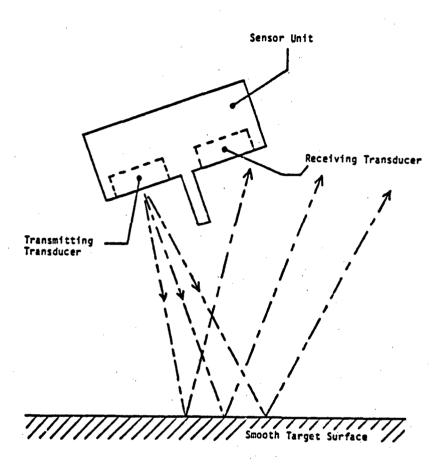
Figure 4.6

Loss of Received Signal Due to Sensor Angle, Under Conditions
of Specular Reflection.

70

4.2.2 Sensor Height Sensitivity

In addition to being angle sensitive, the sensor is also height sensitive. Figure 4.7 illustrates this height sensitivity; as in Figure 4.5, the peak voltage value of the receiver's demodulator output is plotted as a function of horizontal sensor position. Note that the peak signal strength is a nonlinear function of sensor height and that even at relatively large heights this peak value is well above the demodulator noise level. A discussion of this nonlinearity is given below.
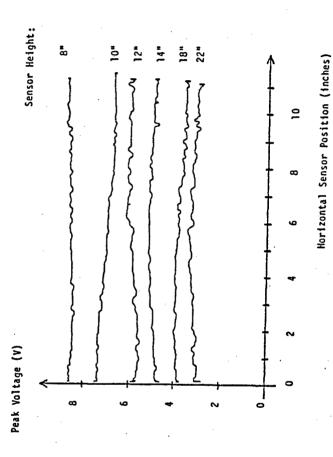
Given a small, fixed-area target, theoretical considerations dictate that the power of the received signal varies inversely with the fourth power of target separation distance, d [15]:

$$\text{received power} = k_1 \left[\frac{e^{-2ad}}{d^4} \right] \times \text{transmitted power} , \qquad (4.1)$$

where a is the attenuation constant of 40 kHz ultrasound in air and k_1 is a constant of proportionality. However, in the above experiment the area of the metal plate target was much greater than that of the transmitted beam, and hence for these purposes may be treated as infinite. In that case, theory predicts that the received signal power varies inversely as only the square of target separation distance:

$$\text{received power} = k_1 \left[\frac{e^{-2ad}}{d^2} \right] \times \text{transmitted power} . \qquad (4.2)$$

71

Target Surface: Metal Plate

Sensor Angle: 0°

Sensor Height:

Peak Voltage (V)

Horizontal Sensor Position (inches)

Figure 4.7

Peak Demodulator Output Voltage as a Function of Horizontal Sensor Position and Height.

72

At small target separation distances, the attenuation may be neglected:

$$\frac{\text{received power}}{\text{transmitted power}} = \frac{k_1}{d^2} \;. \tag{4.3}$$

For a piezoelectric transducer, the voltage generated by incident acoustic power, P, is

$$\text{generated voltage} = k_2\sqrt{P} \;. \tag{4.4}$$

Taking the square root of both sides of equation 4.3 and substituting into equation 4.4 yields an inverse relation between the generated transducer voltage as a function of sensor height:

$$\text{generated voltage} = \frac{k_3}{d} \;. \tag{4.5}$$

Equation 4.5 was verified by plotting average values of the experimental peak voltage data of Figure 4.7 as a function of (1/d), the reciprocal of sensor height. This plot is shown in Figure 4.8.

Returning to the comparison of sensor height and angle sensitivity as discussed at the beginning of this section, further comparisons of Figures 4.5 and 4.7 indicate that the sensor is much less sensitive to linear displacements than to angular displacements. This is a useful observation, for it indicates that the sensor may be used with confidence at relatively large target ranges provided that the sensor angle is small, i.e., less than approximately five degrees.

4.2.3 Environmental Sensitivity of the Ultrasonic Rangefinder

The sensor's angle sensitivity to smooth surfaces, as noted in

73
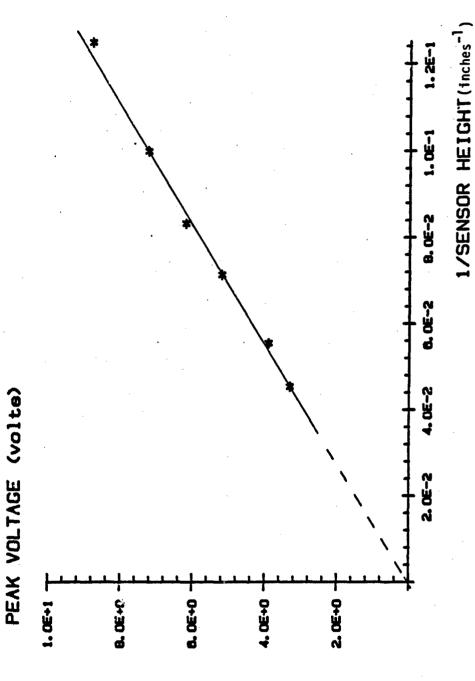
Figure 4.8

Demodulator Peak Output Voltage as a Function of (1/d), the Reciprocal of Sensor Height.

74
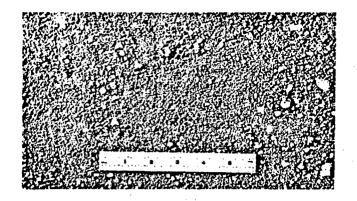
section 4.2.1, is due to the specular mode of reflection which is occurring at the target. As the target surface roughness is increased, the incident ultrasound will be scattered and only a portion of this energy will be detected by the receiver. To demonstrate this phenomenon, peak signal strength was plotted for five different surfaces: dirt (a), sand (b), small gravel (c), large gravel (d), and large stones (e). See Figure 4.9 for an illustration of these surfaces. In this experiment, the sensor's height above surfaces (a)-(e) was fixed at eight inches while the sensor angle was fixed at zero degrees. As before, the sensor was horizontally swept over the target surfaces. Note from Figures 4.10-4.14 that the peak signal strength varies greatly as the sensor is horizontally moved. These variations, much larger than those encountered with the metal plate target of Figure 4.7, are to be expected, due to the gross inhomogeneity and roughness of the target surfaces. This data suggests that the system's ability to gage target range over rough surfaces may be marginal at best, due to the frequent dropouts in received signal. A possible solution to this dropout problem may be to greatly increase the transmitted power level. Thus, when diffuse reflection occurs, the relatively small percentage of received signal will still be sufficient to accurately trigger the receiver circuitry. Another possibility would be to vary the frequency of the transmitted ultrasound, as in the Polaroid Ultrasonic Rangefinder. This technique, as discussed in section 2.6.1, reportedly avoids dropouts associated with any single frequency.
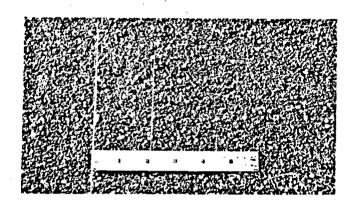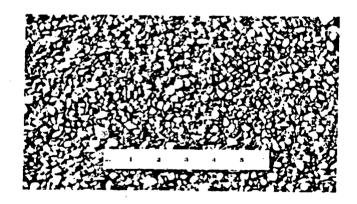
Again, with diffuse reflection, incident ultrasound is scattered after striking a relatively rough target surface. Hence, at relatively

75
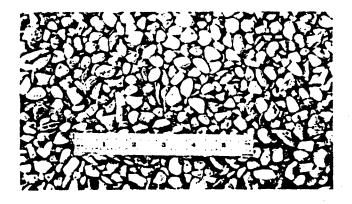
(a) Dirt

(b) Sand

(c) Small Gravel
Figure 4.9
Target Surfaces Used to Test the Rangefinder's Environmental Sensitivity.

76
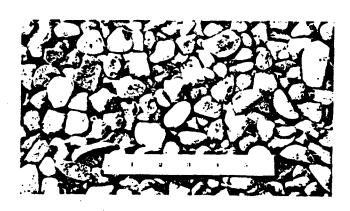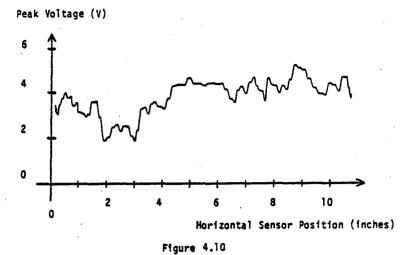
(d) Large Gravel

(e) Large Stones

Figure 4.9 (cont'd)

77

Peak Voltage (V)

Horizontal Sensor Position (inches)

Figure 4.10

Peak Signal Strength of Demodulator Output as a Function of
Horizontal Sensor Position, for Target (a), Dirt .

Peak Voltage (v)

Horizontal Sensor Position (inches)
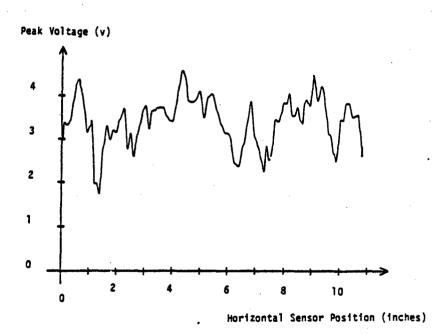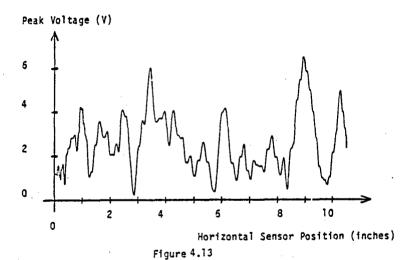
Figure 4.11
Peak Signal Strength, Target (b), Sand.

78

Peak Voltage (V)

Figure 4.12

Peak Signal Strength, Target (c), Small Gravel.

Peak Voltage (V)

Horizontal Sensor Position (inches)

Figure 4.13

Peak Signal Strength, Target (d), Large Gravel.

79

Peak Voltage (V)

Horizontal Sensor Position (inches)
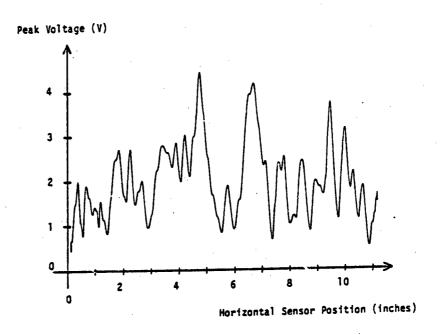
Figure 4.14

Peak Signal Strength, Target (e), Large Stones.

large sensor angles, the percentage of received signal is substantially greater than that encountered when the surface is smooth and specular reflection occurs. Figures 4.15-4.18 demonstrate this fact using a target surface of small gravel, surface (c). Note that the average signal strength is reduced as the sensor angle is increased, but at a much slower rate than that encountered in Figure 4.5 where the target surface was smooth. From Figures 4.15 and 4.18, a sensor angle of fifteen degrees yields an average value of signal strength that is nearly 65% of its zero-angle value. This relative increase in the percentage of received signal, as compared to the specular reflection data, is important, for it indicates that larger sensor angles are tolerable when diffuse reflection occurs.

4.3 Transmitter/Receiver Electronics

In this section the operation of the analog circuitry is discussed. To provide a fuller understanding of its operation, photographs of typical oscilloscope waveforms at various points in the transmitter and receiver circuits are shown. In the transmitter circuit, Figure 4.19 shows the relationship between the digital START signal and the monostable multivibrator output; Figure 4.20 shows the outputs of the oscillator and power booster circuits. A magnified view of the ringing present at the transmitting transducer is shown in Figure 4.21. The falling edge of the top trace of Figure 4.21 indicates the time at which the transducer is turned off. Typical receiver waveforms are shown in Figures 4.22 and 4.23 when a smooth metal plate is positioned parallel to and five inches in front of the sensor face. Figure 4.22 shows the outputs of the preamplifier and active bandpass filter.

81

Peak Voltage (V)

Horizontal Sensor Position (inches)
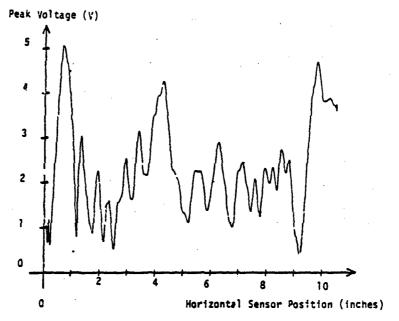
Figure 4.15

Peak Signal Strength with Small Gravel, Sensor Angle = 0°.

Peak Voltage (V)

Horizontal Sensor Position (inches)
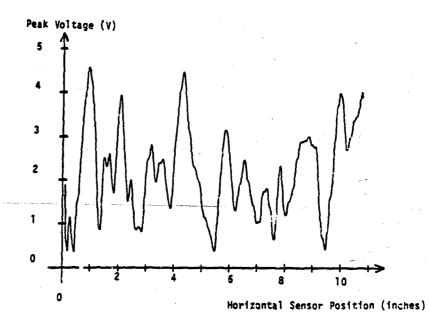
Figure 4.16

Peak Signal Strength with Small Gravel, Sensor Angle = 5°.

82

Peak Voltage (V)

Horizontal Sensor Position (inches)
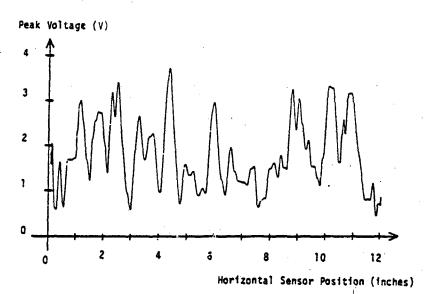
Figure 4.17

Peak Signal Strength with Small Gravel, Sensor Angle = 10°.

Peak Voltage (V)

Horizontal Sensor Position (inches)
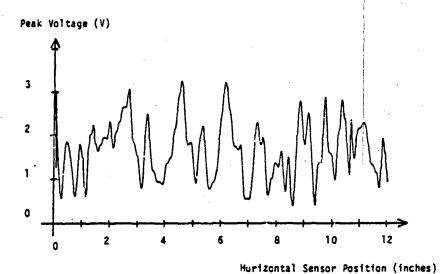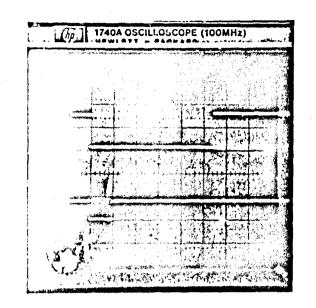
Figure 4.18

Peak Signal Strength with Small Gravel, Sensor Angle = 15°.

83

START
Input

2 V/div

1 mS/div

Monostable
Multivibrator
Output

5 V/div

Figure 4.19

START and Monostable Multivibrator Output Timing.

Oscillator
Output

10 V/div

0.5 mS/div

Power Booster
Output

10 V/div

Figure 4.20

Oscillator and Power Booster Outputs.

84

Gating
Signal

5 V/div

20 µS/div

5 V/div
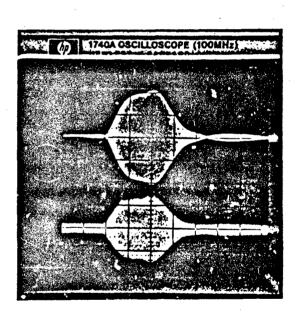
Transducer
Output

Figure 4.21

Transmitting Transducer Ringing Phenomenon Immediately Following
Turn-Off.

Bandpass Filter
Output

2 V/div

0.5 mS/div

0.2 V/div
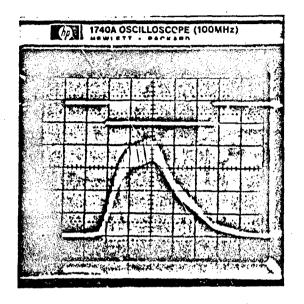
Pre-Amplifier
Output

Figure 4.22

Pre-Amplifier and Bandpass Filter Outputs.

85

Demodulator

5 V/div

0.5 mS/div

1 V/div

Figure 4.23

Digital STOP Output Timing as a Function of Demodulator Output Voltage.

Figure 4.23 shows the outputs of the demodulator and the corresponding timing of the TTL-level digital STOP signal.

As breadboarded, the analog electronics exhibit two adverse characteristics which do not impair the overall performance of the sensor system, but which are discussed here to aid in the possible redesign of future sensor systems. The first is a consequence of the close proximity of and the lack of isolation between the receiver circuits on the analog circuit board. It was experimentally found that strong signals present in one receiver could induce similar signals into adjacent receivers, thus falsely triggering their STOP outputs. As mentioned above, this coupling does not affect the performance of the overall system, due to the time-multiplexing of the STOP signals in the digital circuit (see Figure 3.11). Nevertheless, this coupling may be eliminated by physically separating or shielding each receiver.

A second undesirable aspect of the analog circuitry is a coupling phenomenon that exists between the transmitters and receivers. This coupling, found to be both electrical and acoustical in nature, is responsible for generating positive and negative reinforcement points (nodes and antinodes) at the demodulator outputs as their respective sensor heights are varied. These nodes were found to exist at every half-wavelength increment in sensor height. The electrical coupling is generated by a leakage from the transmitters' 40 kHz oscillator output into the receiver circuits, and may be eliminated by isolating the transmitter circuitry from the receiver circuitry. A second, much weaker, coupling was found to exist when a transmitter and receiver were totally isolated on separate breadboards and powered with separate

87

power supplies. This coupling, experimentally verified to be acoustical in nature, was found to generate nodes and antinodes as before, at every half-wavelength change in vertical sensor height. This acoustical coupling exists between the transmit and receive transducers, and is easily reduced by surrounding the back of each transducer with sound-absorbing material.

Chapter 5

APPLICATION OF THE 40 KHZ SENSOR SYSTEM TO
THE OSU HEXAPOD

The application of the 40 kHz sensor system to the OSU Hexapod is discussed in this chapter. This application includes both the sensor hardware mounting and the development of algorithms used in the control of Hexapod foot altitudes. For clarity, an overview of the existing Hexapod control scheme is also discussed.

5.1 Sensor Hardware Mounting Configuration

The analog and digital electronics have been housed in a metal enclosure and mounted to the Hexapod frame; see Figure 5.1. Data link and power supply connections are visible at the rear of the enclosure; twelve BNC connectors, two per sensor, are visible on the underside of the enclosure and are shown in Figure 5.2. Small diameter coaxial cable, used to minimize electrical interference from external sources, connects each transducer with its corresponding analog circuitry.

The sensors are mounted twelve inches from the tip of each foot and are forward looking. The selection of the sensor mounting height was chosen to satisfy the rangefinder's minimum range of approximately four inches and to prevent a mounting difficulty with the force

89
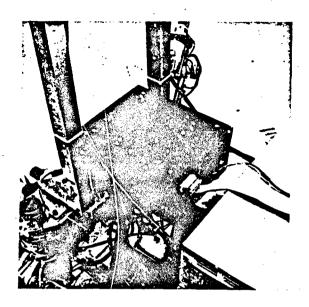
Figure 5.1

40 kHz Sensor Electronics Mounting on the OSU Hexapod.
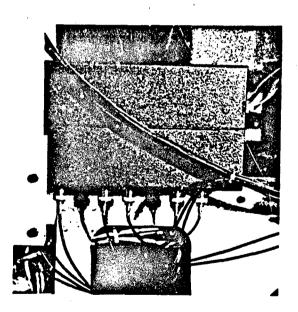
Figure 5.2

Underside View of Sensor Electronics Enclosure Showing
Ultrasonic Sensor Cable Connections.

90

transducer electronics which are housed inside the Hexapod's lower leg segments. The forward-looking sensor mounting configuration, as shown in Figure 5.3, was chosen so that the sensor could anticipate obstacles before the leg actually contacts or moves over them. Of course, this configuration severely limits the system's sensing effectiveness when the Hexapod maneuvers in the reverse direction.

5.2 Overview of Existing Hexapod Control Software

To aid in the understanding of the software developed for the control of Hexapod foot altitudes, a brief overview of the existing Hexapod control scheme is first presented. In this thesis, all work has been done using Hexapod Control Software Version 4.0 [27]. Currently under development, this restructured control scheme has been upgraded from Version 3.5 [6] to permit six-degree-of-freedom body control. Changes notwithstanding, the overall Hexapod control philosophy remains intact, and is discussed below for clarity.

Hexapod control software is currently partitioned into a set of well-defined functional blocks, or subtasks [6]. This control task partitioning is shown in Figure 5.4; the directional flow of information between subtasks is indicated by the arrows. Although it is beyond the scope of this work to justify and analyse each subtask, a brief description of each functional block is clearly in order.

The executive software provides the interface with the human operator, who specifies via a CRT terminal desired vehicle parameters such as velocity, direction, and operating mode. The body motion planning routine preconditions these operator inputs for use by the lower-level subtasks. The primary function of the leg coordination

91
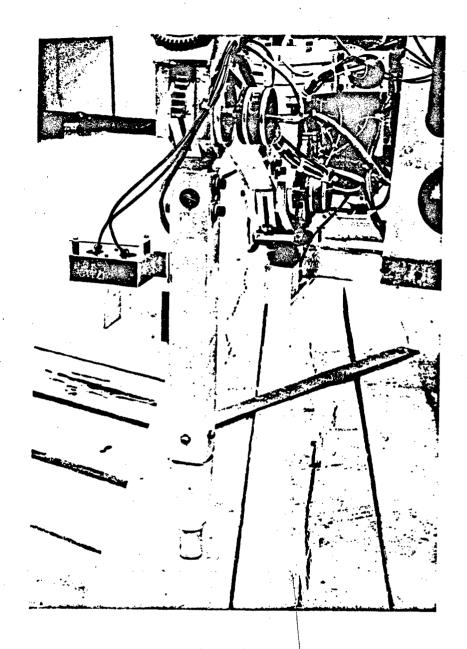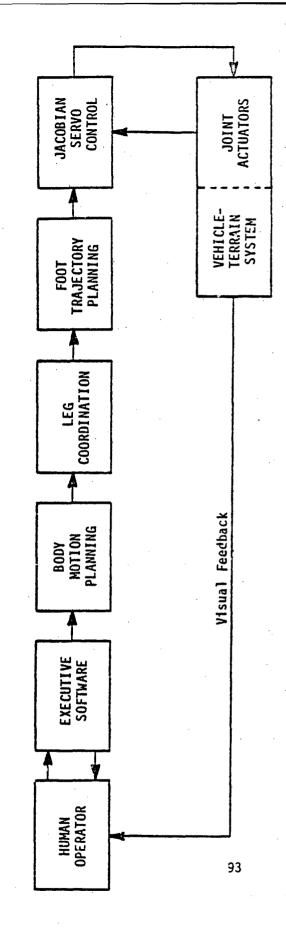
Figure 5.3

Forward-Looking Mounting Configuration of the 40 kHz Sensor Units.

92

Figure 5.4

Hexapod Control Software Task Partitioning.

93

software is to insure that the Hexapod is always supported by a stable set of three or more legs, and that during locomotion, the legs do not collide with one another or with the vehicle body. The foot trajectory planning subtask specifies how the legs are to be moved during Hexapod locomotion in order to realize the operator commands obtained via the executive software. Current Hexapod foot trajectory software allows for several modes of locomotion, including a cruise mode, a side-step mode, and a turn-in-place mode [4]. Finally, the Jacobian servo control subtask is responsible for converting commanded foot trajectories into actual Hexapod joint positions and rates. A detailed description of the latter subtask is given in the following section.

5.2.1 Review of Hexapod Jacobian Servo Control

A block diagram of the Jacobian control structure for one leg of the OSU Hexapod is shown in Figure 5.5 [6]; a description of each servo parameter may be found in Table 5.1. In this error-driven servo, a velocity loop closed in angular joint coordinates controls the angular velocity of a given joint actuator, while a position loop closed in rectangular body coordinates controls the position and velocity of a particular foot tip. Each leg servo has two inputs: a desired position, $\underline{X}_d$, and a desired rate, $\underline{\dot{X}}_d$. Commanded foot rates, $\underline{\dot{X}}_c$, are generated by errors in either foot position or velocity. The inverse Jacobian matrix [26] converts these rectilinear rate commands into angular joint rate commands, $\underline{\dot{\theta}}_c$. If differences exist in the actual and commanded joint rates, an error voltage, $\underline{V}$, is generated which tends to force these differences to zero.
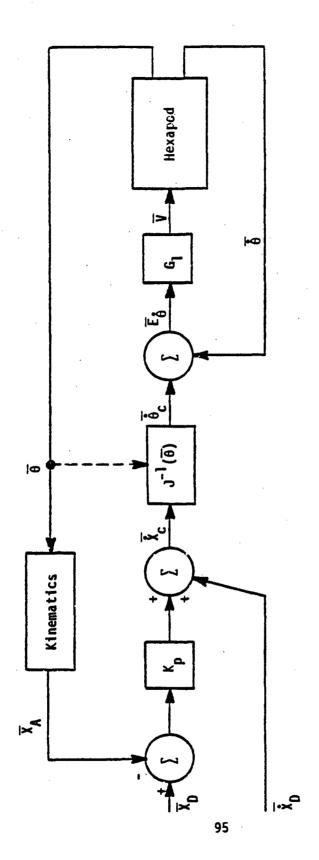
Figure 5.5

Jacobian Control Structure for One Leg of the OSU Hexapod.

Table 5.1

JACOBIAN CONTROL PARAMETER DEFINITIONS

$\overline{X}_D \equiv [X_D \ Y_D \ Z_D]^T$ (desired foot position expressed in body coordinates)

$\overline{X}_A \equiv [X_A \ Y_A \ Z_A]^T$ (actual foot position expressed in body coordinates)

$\overline{\dot{X}}_D \equiv [\dot{X}_D \ \dot{Y}_D \ \dot{Z}_D]^T$ (desired foot velocity expressed in body coordinates)

$\overline{\dot{X}}_A \equiv [\dot{X}_A \ \dot{Y}_A \ \dot{Z}_A]^T$ (actual foot velocity expressed in body coordinates)

$\overline{\dot{X}}_C \equiv [\dot{X}_C \ \dot{Y}_C \ \dot{Z}_C]^T$ (commanded foot velocity expressed in body coordinates)

$\overline{\dot{\theta}}_C \equiv [\dot{\psi}_C \ \dot{\theta}_{1C} \ \dot{\theta}_{2C}]^T$ (vector of joint rate commands)

$\overline{E}_{\dot{\theta}} \equiv [e_\psi \ e_{\theta 1} \ e_{\theta 2}]^T$ (vector of joint rate errors)

$\overline{V} \equiv [V_\psi \ V_{\theta 1} \ V_{\theta 2}]^T$ (vector of joint actuator input voltages)

$\overline{\theta} \equiv [\psi \ \theta_1 \ \theta_2]^T$ (vector of actual joint angles)

$\overline{\dot{\theta}} \equiv [\dot{\psi} \ \dot{\theta}_1 \ \dot{\theta}_2]^T$ (vector of actual joint rates)

$$K_p \equiv \begin{bmatrix} k_{pX} & 0 & 0 \\ 0 & k_{pY} & 0 \\ 0 & 0 & k_{pZ} \end{bmatrix}$$ (position gain matrix)

$J(\overline{\theta}) \equiv$ Jacobian matrix which converts from joint rates to rectilinear foot velocity

Kinematics $\equiv$ equations which convert joint angles to rectilinear foot coordinates

96

5.2.2 Review of Hexapod Leg Motion During Locomotion

During Hexapod locomotion, each leg cycles through two distinct phases: a foot-support phase and a foot-transfer phase. The foot-support phase refers to the time at which a leg is in contact with the ground and actively supporting weight, as determined by foot-contact force sensors [6]. The foot-transfer phase denotes the time at which a leg is off of the ground and moving to a new position, in preparation for its next support phase.

In Hexapod Control Software Version 4.0, the foot-transfer phase is divided into three separate motions, or subphases: foot liftoff, foot return, and foot placedown. In the foot-liftoff subphase the foot is vertically lifted a constant height above the ground, at which time the return subphase begins. During this period the foot altitude is fixed as the leg is moved toward its next support-phase location; at this point the foot is then vertically lowered during the foot-placedown subphase. It is during the two latter subphases of foot transfer that Hexapod foot altitudes will be modified under the control of proximity sensor data. For simplicity, in all experiments a tripod gait is used in which a stable tripod of three legs is in contact with the ground at any given time.

5.3 Proximity Data Acquisition and Conversion Software

The proximity data acquisition and conversion software is the link which connects the sensor hardware of Chapter 3 with the foot altitude control software discussed below. This software is responsible for gathering the range counter output data from the digital circuit and

97

converting it to usable values of Hexapod foot altitudes.

The six channels of range counter output data obtained from the six corresponding data registers in the digital circuit represent the raw, unprocessed Hexapod foot altitude data. This data is collected by the PDP 11/70 via a simple software loop which sequentially selects and then reads each data register. A constant range offset is subtracted from this raw data to compensate for the twelve inch sensor mounting height above each Hexapod foot tip. The value of offset required for each leg is simply the corresponding range counter output when the Hexapod's lower leg segments are perpendicular to and in contact with their respective support surfaces. The results of the subtraction are multiplied by a proportionality factor which accounts for the speed of sound in air and the frequency of the digital circuit's range counter clock. From equation 2.4,

$$\text{foot altitude} \quad = \quad V_{air}(T) \left(\frac{\text{round-trip time}}{2} \right) \quad . \quad (5.1)$$

The round-trip time is derived from the counter output as follows:

$$\text{round-trip time} \quad = \quad \left[\frac{\text{range counter output - offset}}{\text{range counter clock frequency}} \right] \quad . \quad (5.2)$$

Since the speed of sound in air, at temperature T (°C) is

$$V_{air}(T) \quad = \quad 12 \times 1087.14 \sqrt{1 + T(°C)/273} \quad \text{(inches/sec)},$$

$$(5.3)$$

98

at a room temperature of 22 °C and a range counter clock frequency of 70 kHz, equation 5.1 may be written as

$$\text{foot altitude} = 0.097 \times (\text{range counter output - offset}).$$
$$(\text{inches}) \qquad (5.4)$$

An alternative, experimental method of determining the constant of proportionality is to merely calculate the reciprocal of the slope of the range counter data presented in Figure 4.3. From an analysis of this data, the slope of the line was found to be approximately 10.68 counts/inch; its reciprocal yields a proportionality constant of 0.094 inches/count, in excellent agreement with the value derived above.

5.4 Foot Altitude Control Utilizing Proximity Data

In this thesis, foot altitude control has been implemented during the latter two portions of the foot-transfer phase: the foot-return subphase and the foot-placedown subphase. The latter, more important subphase, is discussed first.

5.4.1 Foot Altitude Control During the Foot-Placedown Subphase

Prior to this work, in all previous Hexapod control software versions, the Hexapod exhibited a "stomping" effect while walking, due to the high velocity of each leg upon ground impact. This phenomenon will be presented in the experimental results of Chapter 6. In Hexapod Control Software Version 4.0, the foot-placedown subphase of the walking algorithm has been programmed to generate constant downward foot velocities, relying upon the foot-contact force sensors

99

to determine when a given leg has actually landed.

In this work, foot-impact forces are reduced by generating foot acceleration profiles such that any given foot moves at a maximum velocity at a point approximately halfway through its total estimated downward displacement, and more importantly, at a much slower rate when approaching its support surface. This modified leg motion is implemented with a very simple acceleration profile as shown in Figure 5.6 and consists of two symmetrical constant acceleration segments, one positive and one negative. In Figure 5.6, t_1 denotes the time at which the acceleration is discontinuous. The resulting velocity profile and position trajectory of each foot, also shown in Figure 5.6, is simply the integral of the acceleration and velocity profiles, respectively. For simplicity, in Figures 5.6 and 5.7, positive acceleration, velocity, and position displacement is defined so as to move a given Hexapod foot toward the ground. Also for simplicity, in this thesis, the foot setdown time, $(T_f - T_0)$, was chosen as constant.

The trajectory discussed above is unrealizable when the maximum desired foot velocity, V_s, is greater than the maximum attainable Hexapod foot velocity, V_{max}. When this condition exists, the acceleration profile of Figure 5.6 must be modified so that the desired peak foot velocity will be clipped at $V_s = V_{max}$. Velocity clipping is accomplished by the insertion of a zero-acceleration segment into the above acceleration profile, as shown in Figure 5.7. The resulting velocity profile and position trajectory is also shown. Note that now there are two times at which the acceleration is discontinuous, t_1 and t_2.

Figure 5.8 illustrates the trajectory-planning routine in terms of

100

Acceleration, a(t)

+A

t_o

time, t

t_1

t_f

-A

Velocity, v(t)

V_s

v_o

V_f

t_o

t_1

t_f

Position, P(t)

P_f

P_o

time, t

t_o

t_1

t_f
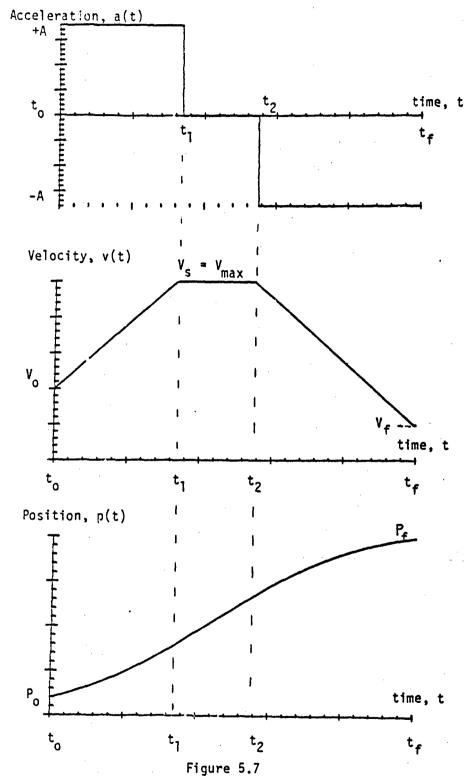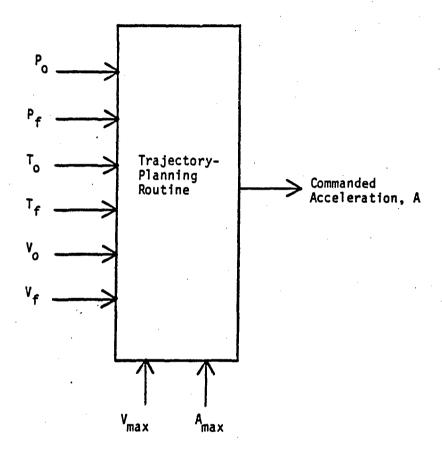
Figure 5.6
Desired Acceleration, Velocity, and Position Profiles
During the Foot-Placedown Subphase.

101

Figure 5.7

Modification of the Trajectory of Figure 5.6 to Include
a Maximum Constant Velocity Period.

Figure 5.8

Input/Output Variables of the Trajectory-Planning Routine
Used in the Control of Hexapod Foot Altitudes During the
Foot-Placedown Subphase.

103

its input and output variables. The routine will generate desired values of vertical foot acceleration necessary to realize one of the aforementioned trajectories, given input values of

P_0 : the initial vertical position of the foot,

P_f : the desired final vertical position of the foot,

T_0 : the initial time at which the foot begins the touchdown subphase,

T_f : the desired foot-touchdown time,

V_0 : the initial vertical foot velocity,

and V_f : the desired final vertical foot velocity at ground impact.

The trajectory-planning routine also has two constant inputs, V_{max} and A_{max}, which are defined as the absolute maximum values of realizable velocity and acceleration, respectively, of a given Hexapod foot tip.

A complete derivation of the trajectory planning equations, i.e., the calculations of acceleration and the required accelerational switchpoint times, t_1 and t_2, may be found in Appendix B.

The computed value of acceleration over time is limited to a set of constant, symmetrical values; hence, a simple numerical integration (Euler Integration) is sufficient to generate analytically correct, i.e., with no approximation errors, values of commanded foot velocity. Likewise, since foot velocity over time is piecewise-linear, a second numerical integration (Trapezoidal Integration) may be performed over velocity to generate analytically correct values of commanded foot position.

A block diagram of the foot altitude control scheme utilizing foot

altitude data during the foot-placedown subphase is shown in Figure 5.9 for one leg of the OSU Hexapod. All quantities shown in this figure, with the exception of foot height as measured by the proximity sensor, FHPS, are measured with respect to the Hexapod's body-fixed coordinate system. At the beginning of the foot-placedown subphase, the integrators shown are initialized with current values of Hexapod foot velocity and position, respectively. The outputs of these integrators form the updated V_0 and P_0 inputs, respectively, of the trajectory-planning routine. Desired values of foot velocity and position, as opposed to their actual values, were chosen to prevent an additional servo loop from being closed around these variables. Hence, the dynamics of the original Jacobian servo routine are preserved. The final position input of the trajectory-planning routine, P_f, is the filtered sum of the foot height as measured by a given proximity sensor, FHPS, and the foot height as calculated using direct kinematics with actual joint potentiometer data, FHDK. Again, this latter quantity is measured with respect to the Hexapod's body-fixed coordinate system as shown in Figure 5.10. The low-pass filter of Figure 5.9 is used to eliminate spurious noise present in either proximity sensor or joint potentiometer data. As such, P_f may be thought of as a filtered "ground position", whose value during foot placedown ideally remains constant.
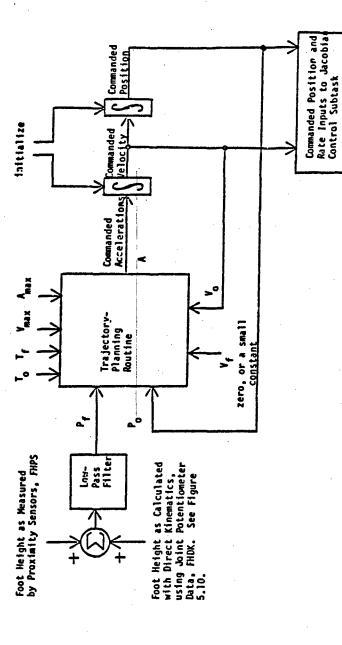
Assuming an ideal servo response, the present desired value of foot position, P_0, will equal the actual foot position as measured with direct kinematics, FHDK. The validity of this assumption is presented in Chapter 6. Since from Figure 5.10 the final value of foot position is given by

105

Figure 5.9

Block Diagram of the Control Structure Used to Implement Foot Altitude Control
During the Foot-Placedown Subphase.
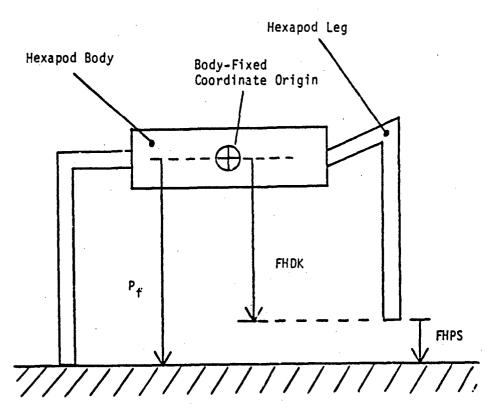
FHPS: Foot Height as measured by Proximity Sensor
FHDK: Foot Height as measured by Direct Kinematics

Figure 5.10

Calculation of a Leg's Final Vertical Placedown Position, P_f.

$$P_f = FHDK + FHPS ,\qquad (5.5)$$

if FHDK = P_0 (ideal servo response), then

$$P_f = P_0 + FHPS .\qquad (5.6)$$

Thus, it is easily seen that the value of vertical foot displacement required to reach the support surface, $(P_f - P_0)$, is determined solely by the foot height as measured by the proximity sensor.

A foot-height measurement problem can exist during the foot-placedown subphase due to the fact that the proximity sensors are not mounted directly underneath the Hexapod feet. As shown in Figure 5.3, the sensors actually detect Hexapod foot height over a point approximately two inches in front of and one inch to the side of any given foot. This configuration presents a difficulty, for example, when the foot is lowered over a stepped surface in which the measured foot height differs from the true foot height. This hazard is illustrated in Figure 5.11; since the proximity data alone ideally determines the required amount of foot displacement, the net result is that the leg is commanded to stop before the foot actually reaches its true support surface. Since the foot-contact force sensor indicates that the leg has not yet reached its support location, i.e., transfer phase not completed, while the trajectory-planning routine is at this time generating zero values of desired foot acceleration, i.e., transfer phase completed, the foot altitude control program effectively "hangs up". To overcome this difficulty, the trajectories as previously discussed are modified to include a constant downward

Descending
Hexapod Leg

Leg-Mounted Sensor Unit

Foot-Contact Force Sensor

Foot Altitude as Measured
by Proximity Sensor

P_f as Determined by
Proximity Sensor

Desired Foot Motion

Desired Support Location

Desired P_f
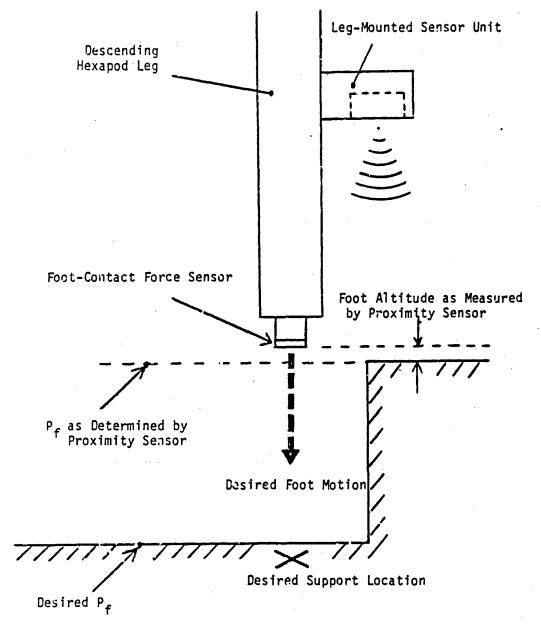
Figure 5.11
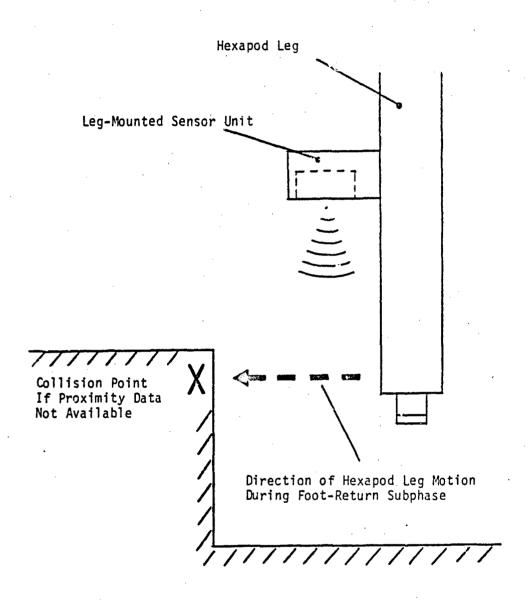
Terrain-Generated Hazard During the Foot-Placedown Subphase,
Due to Ultrasonic Sensor Mounting Configuration.

velocity "search phase" in which foot-contact force sensor data, as opposed to proximity sensor data, is used to terminate the foot-placedown subphase. The search phase is initiated when the proximity sensor data indicates that the leg is within four inches of its touchdown point and when the present commanded foot velocity, V_0, is equal to the desired foot-touchdown velocity. The latter constraint insures that foot velocity remains continuous during the transition to the constant velocity search phase.

5.4.2 Foot Altitude Control During the Foot-Return Subphase

The primary motivation for the control of Hexapod foot altitudes during the foot-return subphase is to prevent the Hexapod from stumbling over obstacles as the feet are moved horizontally toward their new foot-placedown positions. For example, Figure 5.12 depicts a leg laterally approaching a stepped surface whose height is larger than the current vehicle foot height. The leg will collide with the step unless some form of anticipatory proximity sensing is used to raise the foot above the top of the step. As mentioned previously, this anticipatory capability is provided by a forward-looking sensor mounting configuration which permits the sensor unit to detect the step before any contact is made.

It was originally intended in this work that foot altitudes were to be controlled during the entire foot-return subphase. This would include the capability of the foot to track the terrain at a constant altitude, as shown in Figure 5.13. However, during the foot-return subphase, the Hexapod's lower leg segment angle, and hence, sensor angle with respect to the ground, typically varies from approximately

110

Hexapod Leg

Leg-Mounted Sensor Unit

Collision Point
If Proximity Data
Not Available

X

Direction of Hexapod Leg Motion
During Foot-Return Subphase

Figure 5.12

Terrain-Generated Hazard During the Foot-Return Subphase.

111

Direction of Hexapod Leg During Foot Return Subphase

Desired Constant Foot Altitude

Desired Leg Motion

Figure 5.13

Desired Terrain Tracking During the Foot-Return Subphase

112

zero to twenty degrees. On level terrain, this leg angle occurs at an average foot height of eight inches. As discussed in Chapter 4, excessive sensor angle at this height is responsible for a degradation in or total loss of the received sensor signal. Typically, this loss of signal generates erratic foot altitude data which can potentially command the leg to move outside of its kinematic limits. Thus, after experimentation, foot altitude control during portions of the foot-return subphase was deemed impractical, due to the aforementioned shortcomings of the proximity sensor system. A partial solution to this problem, as implemented in this thesis, is described below.

As shown in Figure 5.12, when controlling foot altitudes during the foot-return subphase, the worst possible hazard occurs when the current foot height is less than the obstacle height. However, it is precisely at this time that the sensor is able to accurately determine foot height, in spite of a possibly large sensor angle, due to the very small value of sensor-to-obstacle distance involved. Hence, in this work a directional control scheme is used during the foot-return subphase in which the foot altitude is increased in the event of an encounter with an obstacle, as shown in Figure 5.12, but not decreased when the foot reaches a dropoff. This mode of operation has been selected to prevent erratic vertical leg motion during the foot-return subphase as a result of a loss of sensor signal. The directional control is implemented in software using threshold logic in which the foot is raised only when the sensor indicates that the measured foot height is below a given height threshold. On level terrain during the foot-return subphase, foot altitude is typically eight inches. Hence, the required value of height threshold must be less than this value if

113

proper operation is to occur. When the proximity sensor indicates that the threshold has been crossed, a constant upward velocity is applied to the foot until the sensor indicates that the foot has regained an acceptable amount of clearance, i.e., the threshold value. At this point the commanded upward foot velocity is reduced to zero and the leg continues on in its lateral motion.

Chapter 6

EXPERIMENTAL RESULTS OF THE FOOT ALTITUDE CONTROL SYSTEM

Experimental results of the foot altitude control system as
presented in Chapter 5 will be discussed in this chapter, and will
primarily focus upon the reduction of foot-impact forces during
Hexapod locomotion. For purposes of comparison, vertical leg-force
data will be presented for three separate cases: (1), force data as a
result of using constant downward foot velocities, (2), force data as
a result of implementing parabolic foot trajectories with the
assumption of level terrain and no proximity sensing capabilities, and
(3), force data as a result of implementing parabolic foot
trajectories with the proximity sensing capabilities developed in the
previous chapters.

Due to the present state of incompletion of Hexapod Control
Software Version 4.0, in the following experiments active compliance
[6] has not been utilized. Also, all data presented in this chapter
has been obtained in real time during Hexapod locomotion by writing
desired vehicle parameters, e.g., leg forces, to magnetic disk. Data
is then retrieved from disk off-line and plotted as desired. Typical
values of the quantity under measurement, as opposed to the use of
best- or worst-case values, are presented to give a realistic

115

indication of the vehicle's behavior in the experiments described
below.

6.1 Vertical Leg-Impact Force Due to Constant Downward Foot Velocities

In order to establish a baseline performance reference, vertical
leg-force data has been recorded during Hexapod locomotion while under
the control of existing Hexapod Control Software Version 4.0. As
mentioned previously, this program generates relatively large
(3-5 inches/second) constant downward foot velocities when in the
foot-placedown subphase, and hence, produces a "stomping" effect while
walking. To provide a fair comparison between the resulting vertical
leg force generated during foot impact as compared to that generated
as a result of using piecewise-linear velocity segments as discussed
in Chapter 5, the average foot velocities of each method were
equated. In each case, this velocity is given by

$$\text{average foot velocity} = \frac{\Delta P}{\Delta t} = \frac{\text{total vertical foot displacement}}{\text{total foot-placedown time}}$$

$$(6.1)$$

When the Hexapod stands on level terrain and in a normalized stance,
i.e., all joint angles zeroed, the total vertical foot displacement
during foot placedown is nominally eight inches. For this
displacement, a foot-trajectory placedown time $(T_f - T_0)$ of 2.5
seconds was chosen so that foot velocity during the foot-placedown
subphase would be clipped at its maximum permissible value of $V_s =$
$V_{max} = 5.0$ inches/second. This value, along with a maximum value of
acceleration, A_{max}, of 10.0 inches/second^2, has generally been

116

accepted as the reasonable performance limit of the OSU Hexapod. The 2.5 second placedown time thus determines the value of constant foot velocity necessary in the first case:

$$\text{required constant leg velocity} = \frac{\Delta P}{\Delta t} = \frac{8 \text{ inches}}{2.5 \text{ seconds}} = 3.2 \frac{\text{inches}}{\text{second}}$$

$$(6.2)$$

Vertical leg-force data has been recorded for the Hexapod while traversing level terrain using constant downward foot velocities of 3.2 inches/second. Figure 6.1 shows the resulting vertical, or Z, force generated by one leg as it strikes the ground and then enters support phase. Note that in this and following diagrams, two complete "steps" are shown for a single Hexapod leg. The first arrow in the figure points to the time at which the foot first contacts the ground, t_c. The second arrow points to the time at which the leg enters its support phase, i.e., becomes one of the three legs in the support tripod. This time is denoted as t_s. If at time t_c the leg had landed softly, no leg force would have been generated, since all Hexapod weight was at that time supported by the alternate tripod of legs. However, note that the actual contacting force in Figure 6.1 at time t_c is not zero, but is in fact almost that of its steady-state support value. This is due to the fact that in the algorithms of Version 4.0, the foot-contact force sensors are used to terminate Hexapod leg motions. Thus, with the foot-impact rates used in this experiment, servo response is such that leg motion ceases only after a time in which a substantial vertical foot force has been developed. Clearly, even with moderate foot-placedown rates of 3.2 inches/second,

117
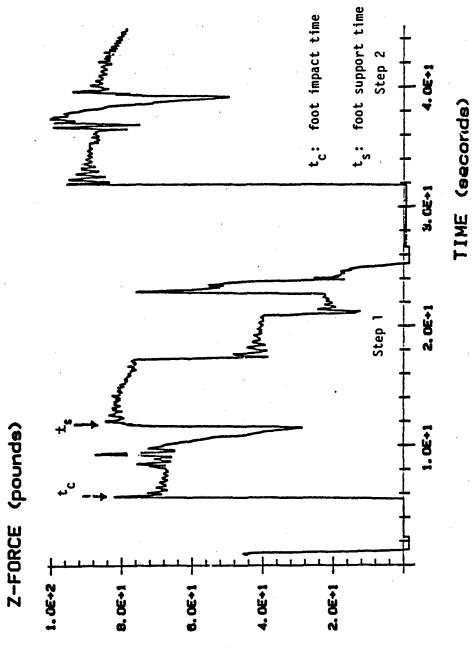
Figure 6.1

Vertical Leg Force Generated by Existing Foot-Placedown Algorithm Using Constant Downward Foot Velocities of 3.2 Inches/Second.

118

foot "stomping" is taking place.

6.2 Vertical Leg-Impact Force as a Result of Implementing Parabolic Foot Trajectories, Assuming Level Terrain and No Proximity Sensing Capabilities

As discussed in Chapter 1, foot altitude control during the foot-placedown subphase could have been implemented on the existing Hexapod if the assumption of level terrain and known body height were made. This possibility is described below. To aid in the discussion, the present body-fixed coordinate system of the OSU Hexapod is shown in Figure 6.2. In this normalized position, the origin resides at a Z height of 18 inches above the ground.

Foot position is controlled with the trajectory-planning routine of Chapter 5 by assigning trajectory input variables such that the desired foot displacement is realized. With a normalized stance on level terrain, these conditions are

$$P_0 = -10.0 \text{ inches,}$$
$$P_f = -18.0 \text{ inches,}$$
$$V_0 = 0.0 \text{ inches/second,}$$
$$V_f = 0.0 \text{ inches/second,}$$

$$\text{and} \quad (T_f - T_0) = 2.5 \text{ seconds.} \tag{6.3}$$

Recall from the previous section that the total leg displacement is nominally eight inches, which was calculated from

$$| P_f - P_0 | = | -18.0 + 10.0 | = 8 \text{ inches.} \tag{6.4}$$

119

Figure 6.2

Existing Hexapod Body-Fixed Coordinate System.
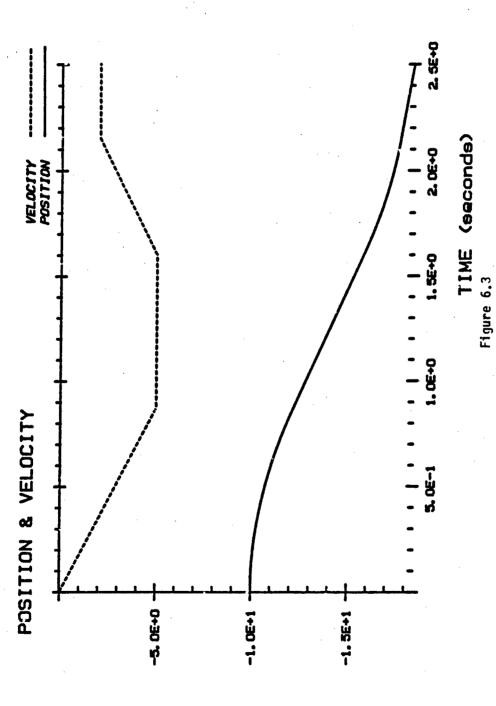
Since the foot begins and ends with zero velocity, ignoring for the present time the constant-velocity search phase, V_0 and V_f are zero. As discussed in the previous section, the foot-placedown time has been chosen as 2.5 seconds. As a result, these input parameters generate desired values of foot position and velocity as shown in Figure 6.3. Note from the figure that velocity clipping occurs midway through the trajectory, and that a constant-velocity search phase of -2.0 inches/second is entered during the latter portion of the foot-placedown routine.

The resulting vertical leg force generated using the above parameters on level terrain is shown in Figure 6.4. Again, the arrows labeled t_c and t_s denote the foot contact and support times of the Hexapod leg, respectively. Note from the figure that the relative magnitude and duration of the initial foot-contact force has been dramatically reduced. This is due to the use of the piecewise-linear velocity profile, which allows us to implement smaller foot rates immediately prior to foot impact, at the expense of greater velocities midway through the trajectory.

6.3 Vertical Leg-Impact Force as a Result of Implementing Parabolic Foot Trajectories With Proximity Sensor Data

The previous method of assuming the vertical location of a foot support surface works well only when the vehicle is on level terrain. Indeed, if an obstacle is present then the foot will typically impact this surface with a higher velocity than that encountered when using constant negative foot velocities of 3.2 inches/second as in section 6.1. This case is shown in Figure 6.5 for the Hexapod traversing an

121

POSITION & VELOCITY

VELOCITY
POSITION

−5.0E+0

−1.0E+1

−1.5E+1

5.0E−1 1.0E+0 1.5E+0 2.0E+0 2.5E+0

TIME (seconds)

Figure 6.3

Desired Foot Position and Velocity, Assuming a Final Foot Position of −18.0 Inches.

122
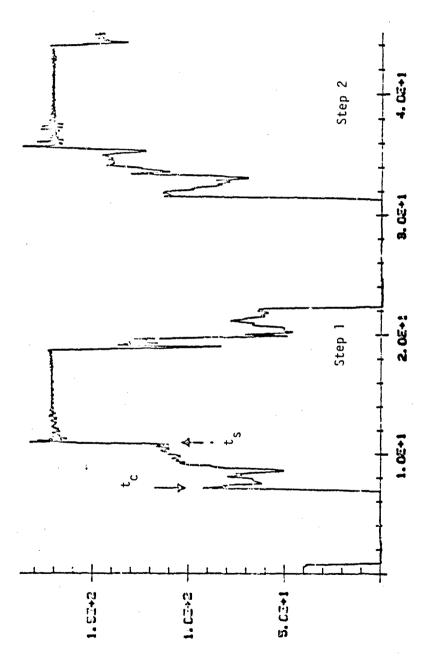
Figure 6.4

Vertical Leg Force Data as a Result of Using a Parabolic Trajectory
On Level Terrain, with No Proximity Sensing Capabilities.

123

Figure 6.5 Vertical Leg-Force Data as a Result of Using a Parabolic Trajectory On Irregular Terrain, with No Proximity Sensing Capabilities.

124

irregular terrain consisting of a set of 4" x 6" wooden beams, as shown in Figure 6.6. The first step in Figure 6.5 pertains to foot placedown on a surface representing level terrain, i.e., the leg's final vertical foot position is equal to that of the other Hexapod legs. The second step pertains to the foot stepping on the aforementioned obstacle. Note that the resulting foot-impact force during the second step is greater than its steady-state value, and much greater than that encountered during level terrain locomotion. This data indicates that this control scheme can easily degrade the performance of Hexapod locomotion over irregular terrain, and dramatically demonstrates the need for a proximity-sensor-based system of foot altitude control during the foot-placedown subphase.
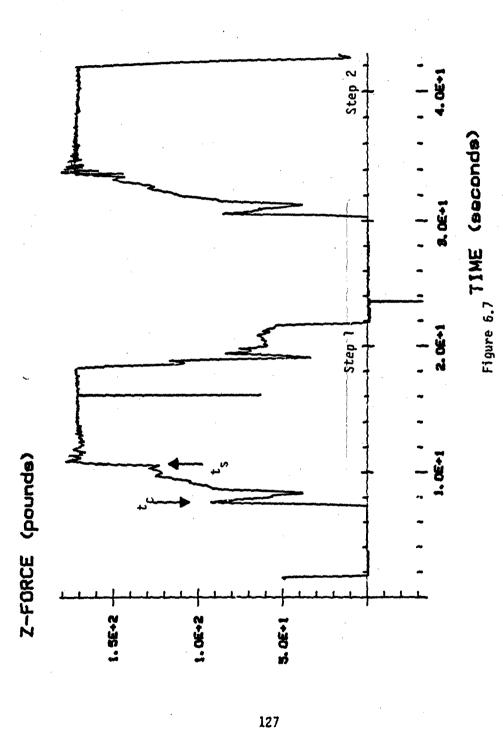
Figure 6.7 shows the resulting leg force for the Hexapod maneuvering over level terrain under the control of proximity sensor data. As expected, the resulting force is very similar in form to that of Figure 6.4 where the vertical position of the level terrain was assumed. The true test of the sensor system is revealed in the second step of Figure 6.8 where the Hexapod has traversed the irregular terrain described above. Note that the resulting leg force while stepping on this obstacle is of the same form as that obtained while traversing level terrain. Hence, foot altitude control during the foot-placedown subphase has been successful in reducing the "stomping" effect present in Hexapod locomotion, even over irregular terrain.

6.4 Servo Response During the Foot-Placedown Subphase

In order to provide an indication of the servo routine's ability

125

Figure 6.6

The OSU Hexapod Traversing Irregular Terrain.

Figure 6.7 Vertical Leg-Force Data as a Result of Using a Parabolic Trajectory with Proximity Sensor Data, On Level Terrain.
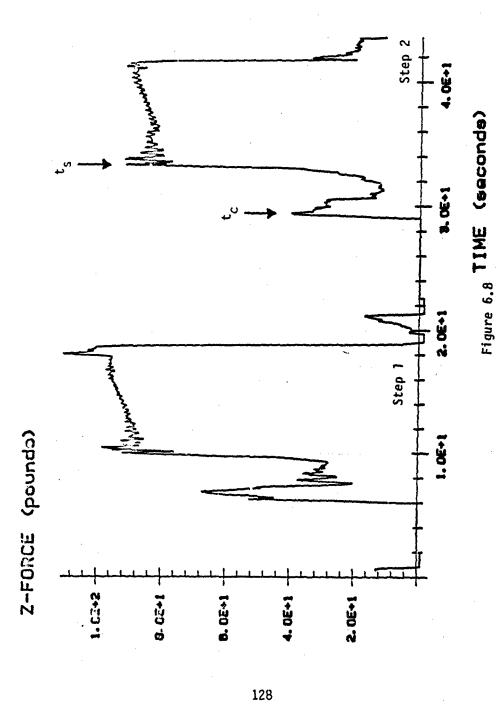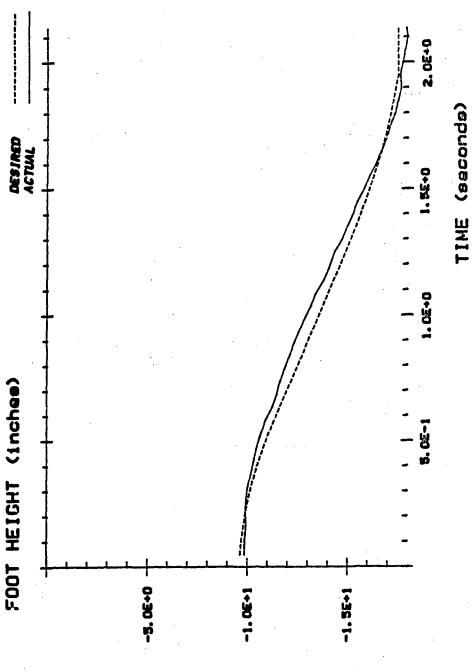
127

Figure 6.8 Vertical Leg-Force Data as a Result of Using a Parabolic Trajectory with Proximity Sensor Data, On Irregular Terrain.

128

to track the desired position and rate commands as generated by the trajectory-planning routine, actual versus desired foot position is plotted as a function of time during the foot-placedown subphase. As shown in Figure 6.9, the actual position during mid-trajectory typically lags its desired value by approximately 0.10 seconds when a trajectory time of 2.5 seconds is used. As such, the amount of lag is small enough that for practical purposes the servo response may be treated as ideal. Of interest in Figure 6.9 are the crossovers in actual and desired foot position. These crossovers are a result of the Simplified Inverse Plant control [28] which has been utilized in the servo routine sections of Hexapod Control Software Version 4.0, and are generated by an overestimation in the velocity required to realize the desired trajectory.

6.5 Results of the Foot Altitude Control System During the Foot-Return Subphase
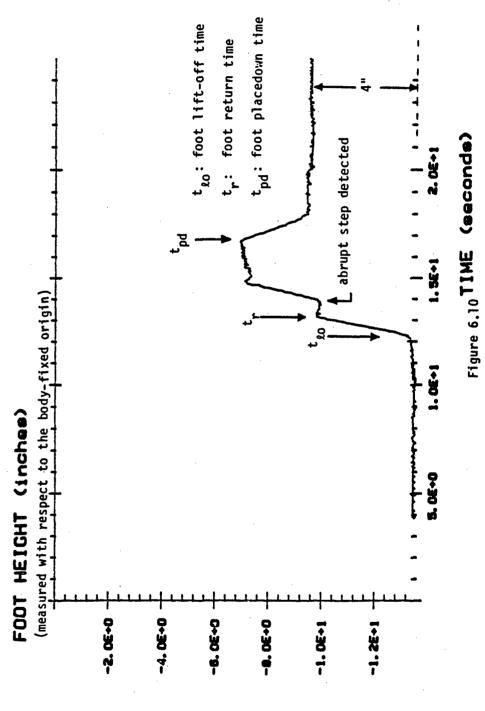
To demonstrate the sensor system's ability to modify foot altitudes during the foot-return subphase, foot height as measured using direct kinematics, FHDK, has been recorded during this time for Hexapod locomotion over the irregular terrain described above. Recall that FHDK is measured with respect to the body-fixed coordinate system of Figure 6.2. In this experiment, the Hexapod has been pre-positioned on the terrain so that an abrupt four inch step is encountered by one leg during its foot-return subphase. The resulting foot altitude data is shown in Figure 6.10. Foot lift-off and return times are denoted as t_{ℓ_0} and t_r, respectively. Note that during the foot-return subphase ($t_r < t < t_{pd}$), the detection of the step generates a positive constant velocity

Figure 6.9

Actual and Desired Foot Height During the Foot-Placedown Subphase.

130

Figure 6.10 Actual Vertical Foot Position During the Foot-Return Subphase, While Traversing Irregular Terrain.

131
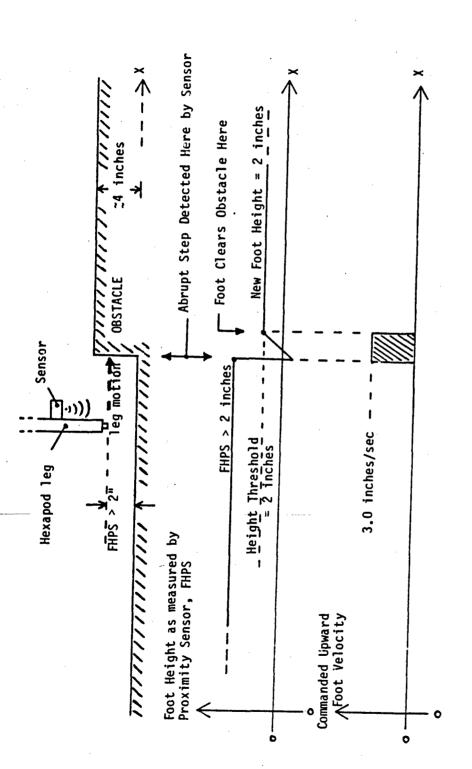
command of 3.0 inches/second which raises the foot so that a collision is avoided. In this experiment, the value of height threshold needed to trigger the upward velocity command is two inches. Hence, this command is withdrawn after the sensor indicates that the new foot height above the surface, i.e., the step, is two inches. For clarity, this procedure is illustrated in Figure 6.11. In Figure 6.10, the time at which the foot enters its placedown subphase is denoted as t_{pd}. Note that the final resting position of the foot is approximately four inches above its original position, corresponding to the four inch thickness of the wooden beam obstacle.

Figure 6.11

Generation of Constant Upward Foot Velocities During the Foot-Return Subphase.

Chapter 7

SUMMARY AND CONCLUSIONS

Foot altitude control for the OSU Hexapod during locomotion over irregular terrain has been implemented in this thesis. Proximity sensor hardware and foot altitude control software has been developed which gives the Hexapod an ability to sense and avoid obstacles and to walk with a much smoother stride. Foot altitude control is applied to each leg during its transfer phase of motion, using proximity sensor data to determine the final support location of each leg. As a result, foot-impact forces have been dramatically reduced both on level and irregular terrain during Hexapod locomotion.

7.1 Research Contributions

The development of a proximity sensor system for the OSU Hexapod vehicle first required an investigation of possible techniques usable in the determination of foot altitudes. Of these techniques, the ultrasonic pulse-echo method of distance measurement was selected, owing to its inherent simplicity and ease of implementation. Several ultrasonic rangefinder designs using this mode of operation were tested and discarded, due to either undesirable operating characteristics or severely limited sensing capabilities.

A custom-designed ultrasonic transmitter and receiver operating at a frequency of 40 kHz was then developed which avoided several of the problems encountered in the previous implementations. Sensor performance has been documented for various values of sensor height and angle above a target, and for five different types of terrain. For relatively smooth target surfaces at reasonable sensor angles and heights, sensor performance has been found to be completely satisfactory. In nine months of operation, the sensor system has also proven to be extremely reliable, with zero failures in either transducers or electronics.

A proximity sensor has been mounted to each Hexapod leg to monitor and control foot altitudes during the foot-return and foot-placedown subphases of each leg's transfer phase. A simple height-threshold control scheme has been implemented during the foot-return subphase in order to avoid obstacles, while a foot trajectory-planning routine is used to specify foot positions and velocities during the foot-placedown subphase, in order to reduce foot-impact forces. It has been demonstrated that this system works equally well while traversing either level or irregular terrain. To the author's knowledge, this capability has never before been demonstrated on a computer-controlled walking machine of any sort.

7.2 Research Extensions

As a result of this work, several modifications and improvements to the existing sensor system are seen to be necessary in order to permit proximity sensor control under most real-world terrain conditions. In terms of hardware, a redesign of the ultrasonic

135

transmitter and receiver circuitry would include the use of multiple transmit frequencies, which should greatly reduce rangefinder sensitivity to terrain characteristics. Also, the inclusion of a variable gain-control circuit in the receiver, in which the gain is increased as the time delay of the received echo is increased, would greatly reduce the sensor system's limitations due to excessive leg angle or height.

The current mounting configuration of one sensor unit on each Hexapod leg is responsible for creating a sensed volume which is biased in the forward direction of Hexapod locomotion. This bias could be reduced or eliminated with an array of sensors positioned on each leg which would collectively gage foot altitudes in each direction of the foot. For example, if moving forward, a forward-looking sensor could be used to anticipate obstacles, while if moving sideways, a side-looking sensor could be used.

An improved rangefinder, able to withstand large values of leg angle and height, would permit an expansion of the foot altitude control scheme presently utilized during the foot-return portion of a leg's transfer phase. For example, full terrain-following capabilites could be implemented so that foot altitude remains constant while maneuvering over irregular terrain.

The foot altitude control software could also be expanded to include a calculation of the minimum time necessary to parabolically lower a foot to the ground. Such calculations, based upon initial and final foot positions and velocities, would permit faster Hexapod locomotion.

APPENDIX A

ADDITIONAL FIGURES

The Model TR-89B/Type 40 is a rugged electroacoustic transducer designed for the efficient generation of ultrasonic energy in air for a wide variety of applications. The one piece housing with integral diaphragm provides a moisture-proof unit, suitable for both indoor and outdoor use when mounted, so that the rear terminals are protected from exposure to the outdoor atmosphere.

Several hundred thousand TR-89B transducers are presently in widespread use in ultrasonic intrusion alarms and other remote control and proximity detection applications.

Performance Characteristics

Figure A.1

40 kHz Transducer Specifications.

138

Outline Dimensions
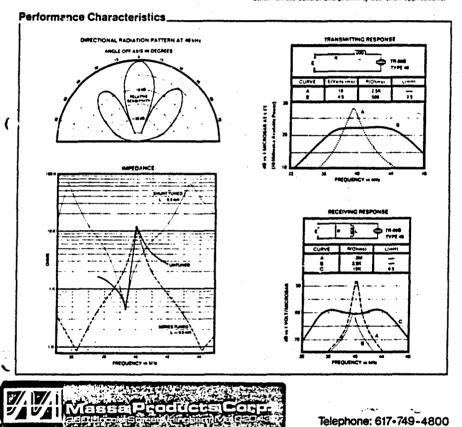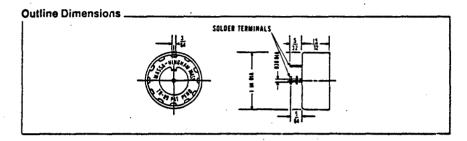
Specifications

FREQUENCY AT MAXIMUM IMPEDANCE (with no load)	40 kHz ± 2 kHz
BANDWIDTH (matched load)	1.0 kHz
TRANSMITTING SENSITIVITY (10 mW available power) (dB vs 1 micro bar at 1 ft.)	+27
RECEIVING SENSITIVITY (untuned with 2MΩ load) (dB vs 1 volt per microbar)	−48
CAPACITANCE AT 1 kHz (nominal)	2300 pF
RECOMMENDED POWER RATING	200 mW
TEMPERATURE COEFFICIENT OF RESONANT FREQUENCY	3 x 10⁻⁴/°F
TUNING INDUCTANCE (nominal)	8.5 mH
WEIGHT	11 grams
STANDARD FINISH	black

Applications

INTRUSION ALARMS, REMOTE CONTROLS, PROXIMITY DETECTION, ECHO RANGING, SOLID OR LIQUID LEVEL MEASUREMENT

Options

1. CLOSER FREQUENCY TOLERANCES
2. DIFFERENT OPERATING FREQUENCIES
3. PHONO CONNECTOR
4. WATER-TIGHT UNIT WITH INTEGRAL CABLE

(AT EXTRA COST SUBJECT TO MINIMUM ORDER REQUIREMENTS)

Patent Notice

MASSA Ultrasonic Transducers are protected by the following U.S. Patents: 2,967,957; 3,128,532; 3,510,698; 3,578,995; 3,638,052; 3,707,131; 3,716,681; 3,752,941; 3,736,632; 3,777,192, and other Patents Pending.

■ 280 LINCOLN STREET, HINGHAM, MASSACHUSETTS 02043 ■ TEL: 617/749-4800

Figure A.1

40 kHz Transducer Specifications, cont'd.

139

Massa Products Corporation

280 Lincoln Street, Hingham, Mass. 02043
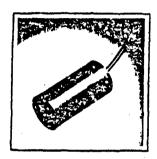
R-283E
Data Sheet
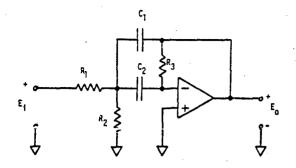
SPECIFICATIONS

Housing Material	ABS Plastic
Sensitivity	
Receiving (dB vs 1V/microbar)	76.5
Transmitting (dB vs 1 microbar/yd/watt)	91.5
Total Beam Width (conical)	
at 3 dB points	13
at 6 dB points	18
at 10 dB points	22
Resonant Frequency (series)	200 kHz
Resistance at Resonance	
(tuned with 0.8 mh choke)	2000 ohms
Power Handling Capacity	
(Maximum peak watts)	100
Dimensions	1½" dia x 3" long
Weight	7 oz.
Cable	Coax. 10 ft.

Over a Generation of Outstanding Leadership in Electroacoustics

Figure A. 2

200 kHz Transducer Specifications.

140

$$\frac{E_0(s)}{E_i(s)} = \left\{ \frac{-\left(\frac{1}{R_1 C_1}\right)s}{s^2 + \left(\frac{C_1 + C_2}{R_3 C_1 C_2}\right)s + \left(\frac{R_1 + R_2}{R_1 R_2 R_3 C_1 C_2}\right)} \right\} = \frac{-H_0 \alpha \omega_0 s}{s^2 + \alpha \omega_0 s + \omega_0^2}$$

where

H_0 = Midband voltage gain,

α = 1/Q,

Q = F_0/BW,

and

ω_0 = $2\pi F_0$.

Solving for R_1, R_2, and R_3 in terms of C_1, C_2, F_0, Q, and H_0 gives

$$R_1 = \frac{Q}{2\pi F_0 H_0 C_1} \quad,$$

$$R_2 = \frac{1}{Q(C_1 + C_2)2\pi F_0 - (1/R_1)} \quad,$$

and

$$R_3 = \frac{Q}{2\pi F_0}\left(\frac{1}{C_1} + \frac{1}{C_2}\right).$$

Figure A.3

Active Bandpass Filter Component Selection.

DERIVATION OF THE TRAJECTORY-PLANNING EQUATIONS

A simple trajectory having symmetrical nonzero acceleration segments is shown in Figure 5.6 and consists of two fitted parabolic segments that are continuous in velocity but not in acceleration. The velocity and acceleration profiles for this trajectory are also indicated in Figure 5.6; note that accelerations A_1 and A_2 during times $(t_0 < t < t_1)$ and $(t_1 < t < t_f)$, respectively, are given by

$$A_1 = \frac{V_s - V_0}{t_1 - t_0} \qquad (B.1)$$

and

$$-A_2 = \frac{V_s - V_f}{t_f - t_1} \qquad . \qquad (B.2)$$

The trajectory accelerations A_1 and A_2 are minimized when their magnitudes are equal; equating magnitudes and solving for the trajectory breakpoint time, t_1, gives

$$t_1 = \left\{ \frac{(V_s - V_0)t_f + (V_s - V_f)t_0}{2V_s - (V_f + V_0)} \right\} . \qquad (B.3)$$

The total change in position is given by the area under the velocity curve of Figure 5.6 and may be written as

$$\Delta P = (P_f - P_0) = \frac{V_0(t_1 - t_0) + V_s(t_f - t_0) + V_f(t_f - t_1)}{2} .$$

$$(B.4)$$

Substituting the value of t_1 from equation B.3 into equation B.4 generates a quadratic equation in terms of V_s:

$$V_s^2 - 2\left[\frac{P_f - P_0}{t_f - t_0}\right]V_s + \left[\frac{(P_f - P_0)(V_f + V_0)}{(t_f - t_0)} - \frac{V_0^2 + V_f^2}{2}\right] = 0$$

$$(B.5)$$

Equation B.3 may also be substituted into equation B.1 to compute the required value of acceleration, A:

$$A = \frac{2V_s - V_f - V_0}{(t_f - t_0)} .$$

$$(B.6)$$

The trajectory breakpoint time, t_1, from Equations B.1 and B.6 is thus

$$t_1 = \left\{\frac{V_s - V_0}{A} + t_0\right\} .$$

$$(B.7)$$

In order to solve for A, and hence t_1, each root of equation B.5 must be tested in equations B.6 and B.7. The root which satisfies

$$t_1 > t_0$$

$$(B.8)$$

is the proper value. If this root dictates a peak value of velocity, V_s, which exceeds the Hexapod's hardware velocity limit, V_{max}, then this trajectory is not realizable with the given input data. Likewise, the

143

trajectory is not realizable if the root dictates a value of required acceleration which is greater than A_{max}, the Hexapod's hardware acceleration limit.

If the peak velocity exceeds the Hexapod's hardware velocity limit while satisfying $A < A_{max}$, a second trajectory having both zero and nonzero acceleration segments may be realized. This trajectory is shown in Figure 5.7 along with its velocity and acceleration profiles. From Figure 5.7, the accelerations A_1 and A_2, during times ($t_0 < t < t_1$) and ($t_2 < t < t_f$), respectively, are

$$A_1 = \frac{V_s - V_0}{t_1 - t_0} \tag{B.12}$$

and

$$-A_2 = \frac{V_s - V_f}{t_f - t_2} \quad . \tag{B.13}$$

Note that there are two unknown trajectory breakpoint times, t_1 and t_2. Equating the magnitudes of accelerations from equations B.12 and B.13 gives

$$t_2 = \left\{ \frac{(V_s - V_f)(t_1 - t_0)}{(V_0 - V_s)} + t_f \right\} \quad . \tag{B.14}$$

Note that in this trajectory V_s equals V_{max}.

The total change in position is again given by the area under the velocity profile and is

$$\Delta P = (P_f - P_0) = \left\{ \frac{V_0(t_1 - t_0) + V_s(t_f + t_2 - t_1 - t_0) + V_f(t_f - t_2)}{2} \right\} \tag{B.15}$$

144

Substituting Equation B.14 into equation B.15 and solving for t_1 gives

$$t_1 = \left\{ \frac{2(P_f - P_0)(V_0 - V_s) + t_0(V_0^2 - 2V_s V_f + V_f^2) + t_f(2V_s^2 - 2V_s V_0)}{(V_0 - V_s)^2 + (V_s - V_f)^2} \right\}$$

(B.16)

The required acceleration is given by substituting this time into Equation B.12:

$$A = \left\{ \frac{(V_0 - V_s)^2 + (V_s - V_f)^2}{2 [V_s(t_f - t_0) - P_f + P_0]} \right\} \cdot$$

(B.17)

If A is greater than A_{max} then this trajectory is not realizable with the given input data.

145

APPENDIX C

DIGITAL CIRCUIT BOARD SCHEMATICS

Note: In the following schematics, IC packages are labled "X/N"
according to the convention that "X" and "N" denote the
column and row position, respectively, of the first pin of
a given package on the digital circuit's wire-wrap breadboard.

IC Locations:

Board Address	Integrated Circuit
A/2	74LS161
A/12	74LS161
A/22	74LS169
A/32	74LS138
C/2	74LS161
C/12	74LS00
C/22	74LS169
C/32	74LS151
E/2	74LS161
E/12	74LS138
E/22	74LS169
E/34	74S37
E/43	74LS109
G/3	74LS161
G/13	74LS161
G/24	10.08 MHz Clock Chip
G/33	74LS30
G/44	74LS30
J/2	74LS10
J/12	74LS109
J/24	74LS08
J/35	74LS32
J/44	74LS32
B1/2	74LS32
B1/11	74LS04

```
B1/20                                      74LS279
B1/30                                      74LS279
B1/44                                      74LS00
D1/2                                       74LS08
D1/11 . . . . . . . . . . . . . . . 74LS08
D1/20                                      74LS00
D1/29                                      74LS00
D1/47                                      HP2630
G1/2                                       74LS378
G1/12 . . . . . . . . . . . . . . . 74LS378
G1/22                                      74LS378
G1/32                                      74LS151
G1/47                                      HP2630
J1/2                                       74LS378
J1/12 . . . . . . . . . . . . . . . 74LS378
J1/22                                      74LS378
J1/32                                      74LS151
J1/47                                      HP2630
B2/2                                       74LS378
B2/12 . . . . . . . . . . . . . . . 74LS378
B2/22                                      74LS378
B2/32                                      74LS151
B2/44                                      MC1488
D2/2                                       74LS378
D2/12 . . . . . . . . . . . . . . . 74LS378
D2/22                                      74LS378
D2/32                                      74LS151
D2/44                                      MC1488
F2/2                                       74LS378
F2/12 . . . . . . . . . . . . . . . 74LS378
F2/22                                      74LS151
F2/32                                      74LS151
F2/44                                      MC1488
H2/2                                       74LS378
H2/12 . . . . . . . . . . . . . . . 74LS378
H2/22                                      74LS151
H2/32                                      74LS151
H2/43                                      74LS151
```

Interface Circuitry Located at PDP 11/70:

```
IF/1                                       HP2630
IF/2                                       HP2630
IF/3                                       HP2630
IF/4                                       HP2630
IF/5                                       HP2630
IF/6                                       MC1488
IF/7                                       MC1488
```

IC Descriptions :

74LS00	Quad 2-Input NAND Gate
74LS04	Hex Inverter
74LS08	Quad 2-Input AND Gate
74LS10	Triple 3-Input NAND Gate
74LS30	8-Input NAND Gate
74LS32	Quad 2-Input OR Gate
74S37	Quad 2-Input NAND Buffer
74LS109	Dual JK Positive Edge-Triggered Flip-Flop
74LS138	1-of-8 Decoder/Demultiplexer
74LS151	8-Input Multiplexer
74LS161	Synchronous Presettable Binary Counter
74LS169	Synchronous Bi-Directional Modulo-16 Binary Counter
74LS279	Quad Set-Reset Latch
74LS378	Parallel D Register with Enable
HP2630	Hewlett-Packard Dual Opto-Isolator
MC1488	Motorola Quad Line Driver
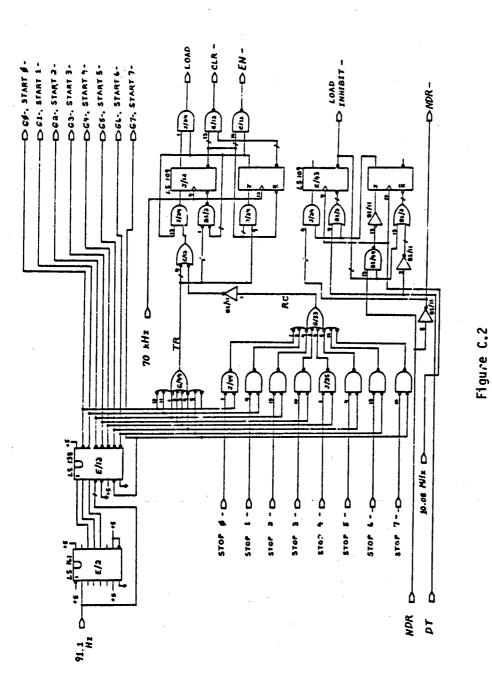
Connectors:

C1-C3	25-pin "D" connector
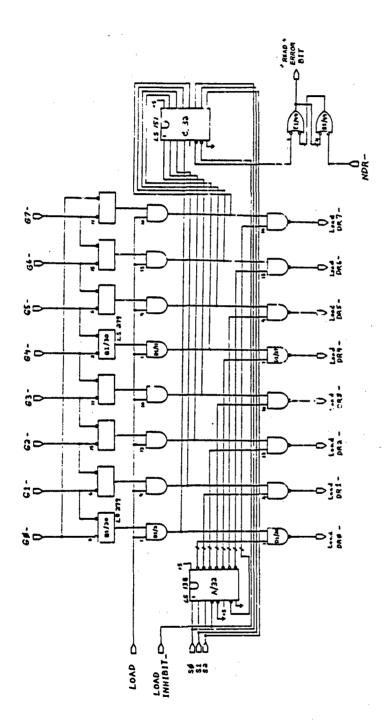
Figure C.1

Clock Generation and Range Counter Circuitry.

Figure C.2

Window Generator, Range Counter Controller, and Load Monitor Circuitry.

150

Figure C.3

Register Loading and Error Detection Circuitry.

Figure C.4

Data Register and Multiplexer Circuitry.

152
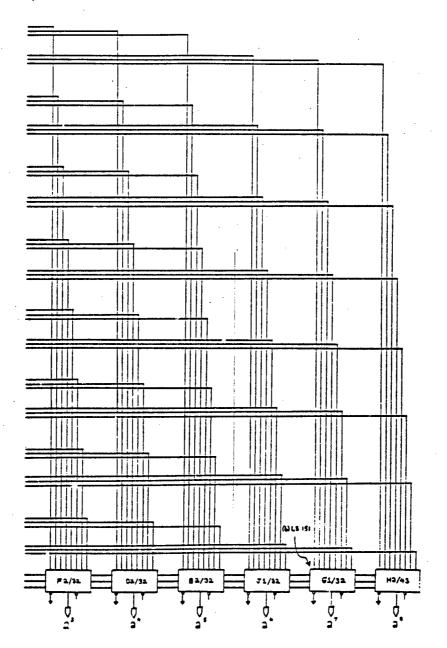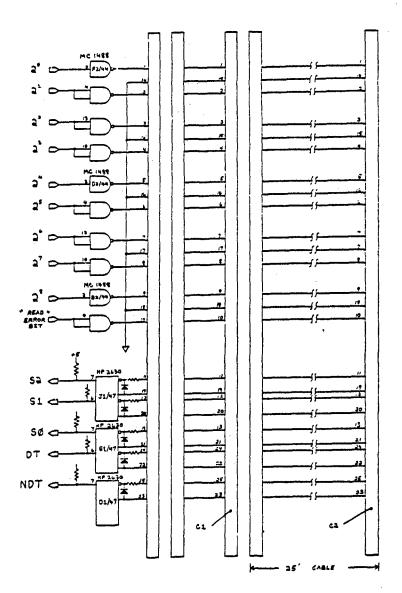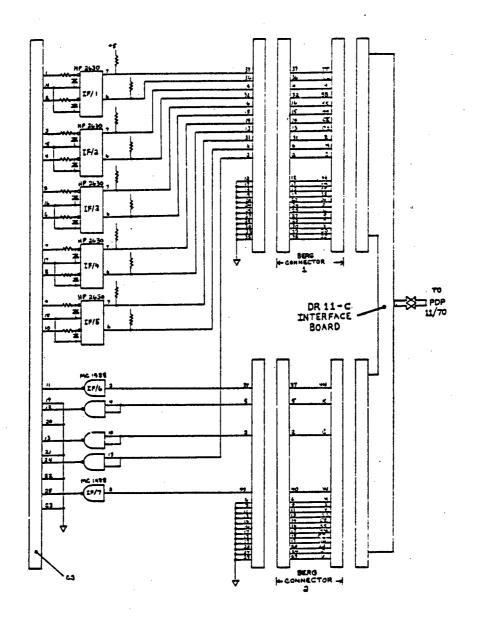
Figure C.4 - continued.

153
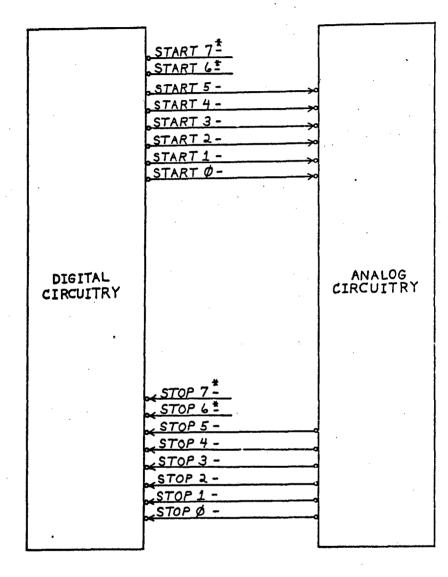
Figure C.5

PDP 11/70 Computer Interface Circuitry.

154

Figure C.5 - continued.

DIGITAL CIRCUITRY		ANALOG CIRCUITRY
START 7 *		
START 6 *		
START 5 -		
START 4 -		
START 3 -		
START 2 -		
START 1 -		
START 0 -		
STOP 7 - *		
STOP 6 - *		
STOP 5 -		
STOP 4 -		
STOP 3 -		
STOP 2 -		
STOP 1 -		
STOP 0 -		

* Not Used

Figure C.6

Analog/Digital Circuit Interconnections.

156

APPENDIX D

ANALOG CIRCUIT BOARD LAYOUT

Power Supply Circuitry:
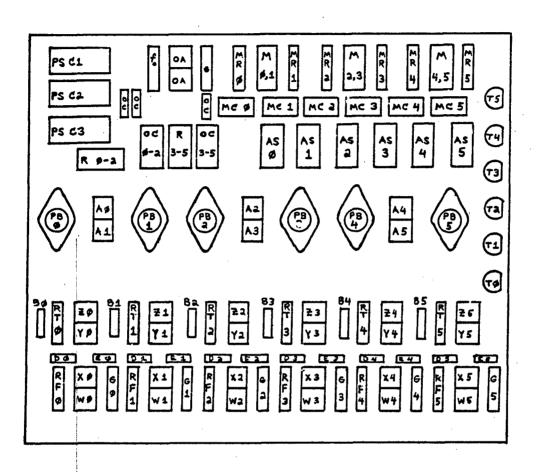
PS C1	+12 V power supply electrolytic capacitor
PS C2	+ 5 V power supply electrolytic capacitor
PS C3	-12 V power supply electrolytic capacitor

Transmitter Circuitry:

OA	40 kHz oscillator circuit, LF351 op-amp
f_0	oscillator frequency adjust potentiometer
G	transmit gain potentiometer
OC	0.001 μF oscillator capacitor
M (0-5)	74LS123 dual monostable multivibrator
MR (0-5)	pulse width adjust potentiometer for M (0-5)
MC (0-5)	monostable multivibrator timing capacitor
OC (0-5)	7406 open-collector inverters
R (0-5)	4.7 kΩ pull-up resistor array
AS (0-5)	AH5010 analog switch
A (0-5)	LF351 power booster pre-amp
PB (0-5)	MC1438R current amplifier

Receiver Circuitry

G (0-5)	receiver gain potentiometer
RT (0-5)	receiver threshold potentiometer
RF (0-5)	receiver bandpass filter frequency adjust potentiometer
B (0-5)	0.032 μF demodulator capacitor
D (0-5)	bandpass filter capacitor
E (0-5)	bandpass filter capacitor
T (0-5)	NPN level-shifting transistor
W (0-5)	preamplifier: LF351 op-amp
X (0-5)	bandpass filter: LF351 op-amp
Y (0-5)	ideal diode: LF351 op-amp
Z (0-5)	threshold comparator: LF351 op-amp

157

Figure D.1

Analog Circuit Board Layout.

158

APPENDIX E

FOOT ALTITUDE CONTROL SOFTWARE

Note: In addition to tripod-gait locomotion under full foot altitude control (command "S"), the main program "SenTest" permits the Hexapod to be normalized (command "N"), or pre-positioned under joystick control via commands "L", "A", or "F".

```
{*********************************************************************}
{****    FILE: TYPE40.PAS :  PASCAL-2  TYPE DECLARATION FILE      ****}
{****                                                            ****}
{****    PROGRAMMER: DENNIS R. PUGH                              ****}
{*********************************************************************}

TYPE    Vector  = RECORD
                      X,Y,Z : Real;
                    END;

        Matrix  = Array[ 1..3 ] of Vector;

        Homogeneous
                = RECORD
                    Rotation    : Matrix;
                    Translation : Vector;
                  END;

        SixVectors  = Array[ 1..6 ] of Vector;
        SixMatrices = Array[ 1..6 ] of Matrix;

        Array3  = Array [0..2] of Real;
        Array6  = Array [1..6] of Real;

        SixBooleans = Array[ 1..6 ] of Boolean;

        LegSet = SET OF 1..6;

        NormalizeMode = ( Ready, Erection, Move1, Move2, Elevation );
        SoftStepMode  = ( ReadyToGo, Step1, Move1st, Step2, Move2nd,
                          Step3, Move3rd, Step4 );

        FootState = ( Support, Transfer );
        SixStates = Array[ 1..6 ] of FootState;

        FootSubstate = ( Ground, LiftOff, Return, PlaceDown, Error );
        SixSubstates = Array[ 1..6 ] of FootSubstate;

        RateType = ( Absolute, Relative );
        SixRateTypes = Array[ 1..6 ] OF RateType;

        Completion = ( NotDone, Done );
        SixCompletions = Array[ 1..6 ] of Completion;

        Coordinate = ( X, Y, Z );
        CoordinateSet = SET of Coordinate;
        SixCoordSets = Array[ 1..6 ] of CoordinateSet;


        AxisControlType = ( PositionControl, ForceControl, BodyControl );
        ControlType = RECORD
                        X, Y, Z : AxisControlType;
                      END;
        SixControlTypes = Array[ 1..6 ] OF ControlType;
```

160

```
String = Packed Array[ 1..10 ] of Char;

TriSetType = RECORD
                    Foot1   : Integer;
                    Foot2   : Integer;
                    Foot3   : Integer;
               END;

TriSetArray = Array[ 1..20 ] OF TriSetType;

YawPairType = RECORD
                    PinnedFoot       : Integer;
                    YawFoot          : Integer;
               END;

YawPairArray = Array[ 1..15 ] OF YawPairType;
```

```
{******************************************************************}
{****    FILE: CONST40.PAS :   CONSTANT DECLARATION FILE        ****}
{****                                                           ****}
{****    PROGRAMMER: DENNIS R. PUGH                             ****}
{******************************************************************}

CONST   Zero = Vector( 0.0, 0.0, 0.0 );
        Identity = Matrix( ( 1.0, 0.0, 0.0 ),
                           ( 0.0, 1.0, 0.0 ),
                           ( 0.0, 0.0, 1.0 ) );

        Pi = 3.14159;
```

```
{***************************************************************}
{****    FILE: EXTERN40.PAS :   EXTERNAL DECLARATION FILE FOR        ****}
{****                          PROCEDURES USED DURING HEXAPOD       ****}
{****                          LOCOMOTION.                          ****}
{****                                                                ****}
{****    PROGRAMMER: DENNIS R. PUGH,                                ****}
{****    MODIFIED BY KEITH R. BROERMAN, JUNE 1, 1983               ****}
{***************************************************************}


PROCEDURE Initialize(
     VAR ActualBodyTransform      : Homogeneous;
     VAR DesiredBodyTransform     : Homogeneous;
     VAR ActualFootPosition       : SixVectors;
     VAR PlannedFootPosition      : SixVectors;
     VAR ActualForce              : SixVectors;
     VAR DesiredForce             : SixVectors;
     VAR Tactilian                : SixMatrices;
     VAR ForceSet                 : SixCoordSets;
     VAR FootContact              : SixBooleans
                          );
EXTERNAL;


PROCEDURE FetchCommand(
     VAR CommandedVehicleMode     : Char;
     VAR LegNumber                : Integer;
     VAR JoyStickDeflection       : Array3;
     VAR WriteFlag                : Boolean
                          );
EXTERNAL;


PROCEDURE LinearMotion(
         JoyStickDeflection       : Array3;
         dt                       : Real;
     VAR PlannedFootRate          : SixVectors;
     VAR FootRateType             : SixRateTypes;
     VAR DesiredBodyLinearRates   : Vector;
     VAR DesiredBodyAngularRates  : Vector;
     VAR DesiredBodyTransform     : Homogeneous
                          );
EXTERNAL;
```

162

```
PROCEDURE AngularMotion(
        JoyStickDeflection        : Array3;
        dt                        : Real;
    VAR PlannedFootRate           : SixVectors;
    VAR FootRateType              : SixRateTypes;
    VAR DesiredBodyLinearRates    : Vector;
    VAR DesiredBodyAngularRates   : Vector;
    VAR DesiredBodyTransform      : Homogeneous
                            );
EXTERNAL;


PROCEDURE LegMotion(
        LegNumber                 : Integer;
        JoyStickDeflection        : Array3;
        dt                        : Real;
    VAR PlannedFootRate           : SixVectors;
    VAR FootRateType              : SixRateTypes;
    VAR DesiredBodyLinearRates    : Vector;
    VAR DesiredBodyAngularRates   : Vector;
    VAR DesiredBodyTransform      : Homogeneous
                            );
EXTERNAL;


PROCEDURE HaltVehicle(
        dt                        : Real;
    VAR PlannedFootRate           : SixVectors;
    VAR FootRateType              : SixRateTypes;
    VAR DesiredBodyLinearRates    : Vector;
    VAR DesiredBodyAngularRates   : Vector;
    VAR DesiredBodyTransform      : Homogeneous
                            );
EXTERNAL;


PROCEDURE NormInitialize;
EXTERNAL;


PROCEDURE NormalizeBody(
    VAR NormalizeBodyFlag         : Completion;
        FootContact               : SixBooleans;
        DesiredFootPosition       : SixVectors;
        dt                        : Real;
    VAR ZeroForce                 : SixVectors;
    VAR PlannedFootRate           : SixVectors;
    VAR FootRateType              : SixRateTypes;
    VAR DesiredBodyLinearRates    : Vector;
    VAR DesiredBodyAngularRates   : Vector;
    VAR DesiredBodyTransform      : Homogeneous
                            );
EXTERNAL;
```

163

```
PROCEDURE SoftInitialize;
EXTERNAL;


PROCEDURE ComputeSensorOffsets(
    VAR Offsets                     : Array6
                          );
EXTERNAL;


PROCEDURE DetermineFootHeight(
        foot                        : Integer;
        Offsets                     : Array6;
    VAR FootHeight                  : Real
              );
EXTERNAL;


PROCEDURE SoftStep(
        FootContact                 : SixBooleans;
        DesiredFootPosition         : SixVectors;
        dt                          : Real;
        DelT                        : Real;
    VAR SoftStepFlag                : Completion;
    VAR ReInitializeFlag            : SixCompletions;
    VAR ActualBodyTransform         : Homogeneous;
    VAR ActualFootPositionBody      : SixVectors;
    VAR ZeroForce                   : SixVectors;
    VAR PlannedFootRate             : SixVectors;
    VAR FootRateType                : SixRateTypes;
    VAR DesiredBodyLinearRates      : Vector;
    VAR DesiredBodyAngularRates     : Vector;
    VAR DesiredBodyTransform        : Homogeneous;
    VAR Offsets                     : Array6;
    VAR LPFOutput                   : Array6;
    VAR POutput                     : Array6;
    VAR VOutput                     : Array6;
    VAR RealTime                    : Real;
    VAR TrajectoryFlag              : SixCompletions;
    VAR FinalTime                   : Array6;
    VAR InitialTime                 : Array6
              );
EXTERNAL;


PROCEDURE GetOrientation(
        Psi                         : Real;    { Yaw angle }
        Delta1                      : Real;    { Pitch gimbal angle }
        Delta2                      : Real;    { Roll gimbal angle }
    VAR Orientation                 : Matrix   { Body orientation matrix }
                          );
EXTERNAL;
```

164

```
PROCEDURE BodyServo(
        DesiredRotationMatrix   : Matrix;  { Desired body orientation }
        ActualRotationMatrix    : Matrix;  { Actual body orientation }
        DesiredBodyAngularRates : Vector;  { Desired angular rates }
        ActualBodyAngularRates  : Vector;  { Actual angular rates }
    VAR BodyServoAngularRates   : Vector   { Servo commanded ang. rates }
                );
EXTERNAL;


PROCEDURE MixFootRates(
        PlannedFootRate         : SixVectors;
        BodyServoFootRate       : SixVectors;
        FootRateType            : SixRateTypes;
        dt                      : Real;
    VAR DesiredFootRate         : SixVectors;
    VAR DesiredFootPosition     : SixVectors
                );
EXTERNAL;


PROCEDURE LegInitialize(
        ActualForce             : SixVectors
                );
EXTERNAL;


PROCEDURE  LegServo(
        DesiredPosition         : SixVectors;
        DesiredForce            : SixVectors;
        DesiredRate             : SixVectors;
        ActualPosition          : SixVectors;
        ActualForce             : SixVectors;
        Tactilian               : SixMatrices;
        ForceSet                : SixCoordSets;
        dt                      : Real;
    VAR CommandRate             : SixVectors
                );
EXTERNAL;


PROCEDURE JointServo(
        CommandRate             : SixVectors;
        ActualRate              : SixVectors;
    VAR OutVoltage              : SixVectors
                );
EXTERNAL;
```

165

```
PROCEDURE Cartesian(
        JointAngles             : SixVectors;
        FootForce               : SixVectors;
    VAR ActualPosition          : SixVectors;
    VAR ActualForce             : SixVectors;
    VAR InverseJacobian         : SixMatrices
                    );
EXTERNAL;


PROCEDURE OutputVolt( Voltage   : SixVectors );
EXTERNAL;


PROCEDURE GyroFeedBack(
    VAR Pitch                   : Real; { Pitch simbal angle }
    VAR Roll                    : Real  { Roll simbal angle }
                    );
EXTERNAL;


PROCEDURE ForceFeedBack(
    VAR FootForce               : SixVectors
                    );
EXTERNAL;


PROCEDURE PositionFeedBack(
    VAR JointAngle              : SixVectors
                    );
EXTERNAL;


PROCEDURE RateFeedBack(
    VAR JointRate               : SixVectors
                    );
EXTERNAL;


PROCEDURE Wait(
    VAR NumberWaitUnits         : Integer;
    VAR WaitUnitType            : Integer;
    VAR WaitStatus              : Integer
                    );
NONPASCAL;


PROCEDURE OffElectronics;
EXTERNAL;


PROCEDURE OnMotors;
EXTERNAL;
```

```
PROCEDURE OffMotors;
EXTERNAL;


PROCEDURE WriteMatrix( OutputMatrix     : Matrix;
                       MatrixLabel      : String );
EXTERNAL;

{***********************************}
{***   LIBRARY ROUTINE SECTION   ***}
{***********************************}


FUNCTION Magnitude( A: Vector ) : Real;
EXTERNAL;


FUNCTION XYMagnitude( A: Vector ) : Real;
EXTERNAL;


PROCEDURE CrossProduct( VAR A: Vector; B, C: Vector );
EXTERNAL;
            { A := B x C }


PROCEDURE VectSub( VAR A: Vector; B, C: Vector );
EXTERNAL;
            { A := B - C }


PROCEDURE VectMult( VAR A: Vector; c: Real; B: Vector );
EXTERNAL;
            { A := c * B }


PROCEDURE VectDiv( VAR A: Vector; B: Vector; c: Real );
EXTERNAL;
            { A := B / c }


PROCEDURE Transpose( VAR At: Matrix; A : Matrix );
EXTERNAL;
            { At := TRANSPOSE( A ) }


PROCEDURE MatrixMult( VAR A: Vector; H: Matrix; B: Vector );
EXTERNAL;
            { A := H * B }
```

167

```
PROCEDURE Transform( Var A: Vector; T: Homogeneous; B: Vector );
EXTERNAL;
                { A := T * B }


PROCEDURE TransIntegrate(    VAR H                : Homogeneous;
                             LinearRate           : Vector;  { wrt. H }
                             AngularRate          : Vector;  { wrt. H }
                             dt                   : Real );
EXTERNAL;
                { H := INTEGRAL( LinearRate, AngularRate, dt }


-------------------------------------------------------------------------

{$own}

{******************************************************************************}
{****     FILE: SENTEST.PAS :   MAIN PROGRAM FOR THE DEMONSTRATION OF   ****}
{****                           FOOT ALTITUDE CONTROL USING PROXIMITY    ****}
{****                           SENSOR DATA.                             ****}
{****                                                                    ****}
{****                           FOR FOOT ALTITUDE CONTROL, ENTER 'S'.    ****}
{****                                                                    ****}
{****     PROGRAMMER: DENNIS R. PUGH,                                    ****}
{****     MODIFIED BY KEITH R. BROERMAN, JUNE 1, 1983                    ****}
{******************************************************************************}


%INCLUDE TYPE40;        { Type declarations }
%INCLUDE CONST40;       { Constant declaration }
%INCLUDE EXTERN40;      { External procedure declarations }


VAR      ActualBodyAngularRates            : Vector;
         DesiredBodyLinearRates            : Vector;
         DesiredBodyAngularRates           : Vector;
         BodyServoAngularRates             : Vector;
         BodyServoAngularRatesEarth        : Vector;
```

168

```
ActualFootPosition                : SixVectors;
DesiredFootPosition               : SixVectors;
ZeroForce                         : SixVectors;
ActualFootForce                   : SixVectors;
ActualJointAngles                 : SixVectors;
ActualJointRates                  : SixVectors;
CommandJointRate                  : SixVectors;
OutVoltage                        : SixVectors;
PlannedFootRate                   : SixVectors;
BodyServoFootRate                 : SixVectors;
DesiredFootRate                   : SixVectors;
LegServoFootRate                  : SixVectors;
ActualForce                       : SixVectors;
DesiredForce                      : SixVectors;

DesiredBodyTransform              : Homogeneous;
ActualBodyTransform               : Homogeneous;

FootRateType                      : SixRateTypes;

WriteFlag                         : Boolean;

FootContact                       : SixBooleans;

InverseJacobian                   : SixMatrices;
Tactilian                         : SixMatrices;

ForceSet                          : SixCoordSets;

JoyStickDeflection                : Array3;

InitialTime                       : Array6;
FinalTime                         : Array6;
Offsets                           : Array6;
LPFOutput                         : Array6;
POutput                           : Array6;
VOutput                           : Array6;

Pitch                             : Real;
Roll                              : Real;
Yaw                               : Real;
CurrentTime                       : Real;
RealTime                          : Real;
LastTime                          : Real;
dt                                : Real;
DelT                              : Real;
T                                 : Real;

foot                              : Integer;
LegNumber                         : Integer;
WaitUnitType                      : Integer;
NumberWaitUnits                   : Integer;
WaitStatus                        : Integer;
```

169

```
                CurrentVehicleMode                    : Char}
                CommandedVehicleMode                  : Char}

                SoftStepFlag                          : Completion}
                NormalizeBodyFlag                     : Completion}

                ReInitializeFlag                      : SixCompletions}
                TrajectoryFlag                        : SixCompletions}


        BEGIN   { SensorTest }

        Initialize(     ActualBodyTransform,
                        DesiredBodyTransform,
                        ActualFootPosition,
                        DesiredFootPosition,
                        ActualForce,
                        DesiredForce,
                        Tactilian,
                        ForceSet,
                        FootContact )}


        NormInitialize}
        SoftInitialize}
        LegInitialize( ActualForce )}
        ComputeSensorOffsets( Offsets )}

        WaitUnitType := 0}      { Select clock ticks for wait units }
        NumberWaitUnits := 2}   { Wait for two clock ticks }

        OnMotors}

        LastTime := Time}
        CurrentVehicleMode := 'H'}
        CommandedVehicleMode := 'H'}

        WriteFlag := FALSE}

        BodyServoAngularRates := Zero}
        Yaw := 0.0}

        NormalizeBodyFlag := NotDone}
        SoftStepFlag := NotDone}
        FOR foot := 1 to 6 DO
           ReInitializeFlag[ foot ] := Done}

        Writeln('Please Enter DelT')}
        Readln(DelT)}
        RealTime := 0.0}
```

170

```
REPEAT { UNTIL CommandedVehicleMode = 'X'}

        CurrentTime := Time;
        dt := ( CurrentTime - LastTime ) * 3600.0;  { seconds }
        LastTime := CurrentTime;
        RealTime := RealTime + dt;

        FetchCommand(   CommandedVehicleMode,
                        LegNumber,
                        JoystickDeflection,
                        Writeflag );


        {*************************************************************}
        {*            OPERATIONAL MODE SEQUENCER SECTION            *}
        {*************************************************************}

        CASE CurrentVehicleMode OF

            'L': CurrentVehicleMode := CommandedVehicleMode;

            'A': CurrentVehicleMode := CommandedVehicleMode;

            'F': CurrentVehicleMode := CommandedVehicleMode;

            'S': IF SoftStepFlag = NotDone
                    THEN
                        CurrentVehicleMode := 'S'
                    ELSE
                        BEGIN
                        CurrentVehicleMode := 'H';
                        CommandedVehicleMode := 'H';
                        END;

            'N': IF NormalizeBodyFlag = Done
                    THEN
                        BEGIN
                        CurrentVehicleMode := 'H';
                        CommandedVehicleMode := 'H';
                        END;

            'H': CurrentVehicleMode := CommandedVehicleMode;

            END; { CASE CurrentVehicleMode }
```

```
{*************************************************************}
{*              REFERENCE GENERATION SECTION               *}
{*************************************************************}

CASE CurrentVehicleMode OF

     'L': LinearMotion(   JoyStickDeflection,
                          dt,
                          PlannedFootRate,
                          FootRateType,
                          DesiredBodyLinearRates,
                          DesiredBodyAngularRates,
                          DesiredBodyTransform );

     'A': AngularMotion(  JoyStickDeflection,
                          dt,
                          PlannedFootRate,
                          FootRateType,
                          DesiredBodyLinearRates,
                          DesiredBodyAngularRates,
                          DesiredBodyTransform );

     'F': LegMotion(      LegNumber,
                          JoyStickDeflection,
                          dt,
                          PlannedFootRate,
                          FootRateType,
                          DesiredBodyLinearRates,
                          DesiredBodyAngularRates,
                          DesiredBodyTransform );

     'S': SoftStep(       FootContact,
                          DesiredFootPosition,
                          dt,
                          DelT,
                          SoftStepFlag,
                          ReInitializeFlag,
                          ActualBodyTransform,
                          ActualFootPosition,
                          ZeroForce,
                          PlannedFootRate,
                          FootRateType,
                          DesiredBodyLinearRates,
                          DesiredBodyAngularRates,
                          DesiredBodyTransform,
                          Offsets,
                          LPFOutput,
                          POutput,
                          VOutput,
                          RealTime,
                          TrajectoryFlag,
                          FinalTime,
                          InitialTime );
```

172

```
'N': NormalizeBody(  NormalizeBodyFlag,
                     FootContact,
                     DesiredFootPosition,
                     dt,
                     ZeroForce,
                     PlannedFootRate,
                     FootRateType,
                     DesiredBodyLinearRates,
                     DesiredBodyAngularRates,
                     DesiredBodyTransform )}


'H': HaltVehicle(    dt,
                     PlannedFootRate,
                     FootRateType,
                     DesiredBodyLinearRates,
                     DesiredBodyAngularRates,
                     DesiredBodyTransform )}


'X': }

END} { CASE CurrentVehicleMode }


OrthoNorm(           DesiredBodyTransform.Rotation,
                     DesiredBodyTransform.Rotation )}


{********************************************************************}
{*                 SERVO ROUTINE SECTION                          *}
{********************************************************************}

GyroFeedBack(  Pitch,
               Roll )}


{ Integrate commanded yaw rate to simulate yaw gyro feedback }
MatrixMult(     BodyServoAngularRatesEarth,
                ActualBodyTransform.Rotation,
                BodyServoAngularRates )}
Yaw := Yaw + BodyServoAngularRatesEarth.Z * dt}

GetOrientation( Yaw,
                Pitch,
                Roll,
                ActualBodyTransform.Rotation )}

IF WriteFlag THEN
        BEGIN
        WriteLn( 'Yaw = ', Yaw:10:4 )}
        WriteLn( 'Pitch = ', Pitch:10:4 )}
        WriteLn( 'Roll = ', Roll:10:4 )}

        WriteMatrix( ActualBodyTransform.Rotation,'ActRotat  ')}
        WriteLn}
        END}
```

173

```
BodyServo( DesiredBodyTransform.Rotation,
           DesiredBodyTransform.Rotation,{for ActualTransform}
           DesiredBodyAngularRates,
           DesiredBodyAngularRates,{for ActualBodyAngularRates}
           BodyServoAngularRates );

FOR foot := 1 TO 6 DO { Convert body rates to foot rates }
        BEGIN
        CrossProduct(    BodyServoFootRate[ foot ],
                         DesiredFootPosition[ foot ],
                         BodyServoAngularRates );

        VectSub(         BodyServoFootRate[ foot ],
                         BodyServoFootRate[ foot ],
                         DesiredBodyLinearRates );
        END;

MixFootRates(    PlannedFootRate,
                 BodyServoFootRate,
                 FootRateType,
                 dt,
                 DesiredFootRate,
                 DesiredFootPosition );

PositionFeedback( ActualJointAngles );

ForceFeedback(   ActualFootForce );

FOR foot := 1 TO 6 DO {subtract force offsets from actual force}
    VectSub(     ActualFootForce[ foot ],
                 ActualFootForce[ foot ],
                 ZeroForce[ foot ] );

Cartesian(       ActualJointAngles,
                 ActualFootForce,
                 ActualFootPosition,
                 ActualForce,
                 InverseJacobian );

IF WriteFlag THEN Write( 'Actual Forces : ' );
FOR foot := 1 TO 6 DO
    BEGIN
    IF -ActualForce[ foot ].Z >= 20.0 { pounds }
        THEN
            FootContact[ foot ] := TRUE
        ELSE
            FootContact[ foot ] := FALSE;

    IF WriteFlag THEN Write( -ActualForce[ foot ].Z:10:2 );
    END;

IF WriteFlag THEN WriteLn;
```

174

```
LegServo(           DesiredFootPosition,
                    DesiredForce,
                    DesiredFootRate,
                    ActualFootPosition,
                    ActualForce,
                    Tactilian,
                    ForceSet,
                    dt,
                    LegServoFootRate );

FOR foot := 1 to 6 DO  { convert foot rates to joint rates }
        MatrixMult(     CommandJointRate[ foot ],
                        InverseJacobian[ foot ],
                        LegServoFootRate[ foot ] );

RateFeedback(   ActualJointRates );

JointServo(     CommandJointRate,
                ActualJointRates,
                OutVoltage );

OutputVolt(     OutVoltage );

{*********************************************************************}
{*                      END OF SERVO ROUTINES                       *}
{*********************************************************************}

WriteFlag := FALSE;

Wait(           NumberWaitUnits,
                WaitUnitType,
                WaitStatus );

IF WaitStatus <> 1 THEN Writeln( 'WaitStatus = ', WaitStatus );

UNTIL CommandedVehicleMode = 'X';

FOR foot := 1 to 6 DO
    OutVoltage[ foot ] := Zero;

OutputVolt( OutVoltage );

OffMotors;
OffElectronics;

END.   { SenTest }
```

175

```
{$nomain}
{$own}

{**********************************************************************}
{****   FILE:  SENSOR40.PAS :   THIS FILE CONTAINS ALL SUBROUTINES    ****}
{****                          USED IN THE CONTROL OF HEXAPOD FOOT    ****}
{****                          ALTITUDES.                            ****}
{****                                                                ****}
{****   PROGRAMMER:  KEITH R. BROERMAN                               ****}
{**********************************************************************}

%INCLUDE TYPE40;        { Type declarations }
%INCLUDE CONST40;       { Constant declaration }
%INCLUDE EXTERN40;      { External procedure declarations }

{*** STATIC VARIABLE DECLARATION ***}

VAR     SoftStepState             : SoftStepMode;
        Step1Flag                 : Completion;
        Step2Flag                 : Completion;
        Step3Flag                 : Completion;
        Step4Flag                 : Completion;
        MoveBodyFlag              : Array[ 1..3 ] OF Completion;
        MoveFootState             : SixSubStates;
        MoveFootFlag              : Array[ 1..6 ] OF Completion;


{**********************************************************************}

PROCEDURE SoftInitialize;

VAR foot        : Integer;

BEGIN

SoftStepState := ReadyToGo;

FOR foot := 1 to 6 DO
    MoveFootState[ foot ] := Ground;

END;    { SoftInitialize }

{**********************************************************************}

PROCEDURE ComputeSensorOffsets;

CONST   Mask = 000777B;

VAR     foot                      : Integer;
        loopcount                 : Integer;
        footminusone              : Integer;

        Select ORIGIN 167742B     : Integer;
        Data   ORIGIN 167744B     : Integer;
```

176

```
          TempOffset                  : Array6;
          Sum                         : Array6;

PROCEDURE Wait;

VAR     Newtime                       : Real;

BEGIN

Newtime := Time;
WHILE (Time - Newtime) * 3600.0 < 0.088  { seconds }  DO  { nothing };

END;

BEGIN           { Determine Senror Offsets }

FOR foot := 1 to 6 DO  Sum[ foot ] := 0;
FOR loopcount := 1 TO 50 DO   { Determine Average Values }
        BEGIN
        FOR foot := 1 to 6 DO
            BEGIN
            footminusone := foot - 1;
            Select := footminusone;
            TempOffset[ foot ] := Data AND Mask;
            Sum[ foot]   := Sum[ foot ] + TempOffset[ foot ];
            END;
        Wait;  { wait 88 milliseconds }
        END;

FOR foot := 1 to 6 DO  Offsets[ foot ] := Sum[ foot ] / 50.0;

END;    { ComputeSensorOffsets }

{******************************************************************}

PROCEDURE DetermineFootHeight;

CONST   Mask = 000777B;
        Sign = 100000B;
        ConversionConstant = 0.094;

VAR     footminusone            : Integer;
        Errorbit                : Integer;

        Select ORIGIN 167742B   : Integer;
        Data   ORIGIN 167744B   : Integer;

        GrossSensorOutput       : Real;
```

177

```
BEGIN   { Determine Foot Altitudes }

footminusone := foot - 1;
Select := footminusone;
Errorbit := Data AND Sign;
GrossSensorOutput := Data AND Mask;
FootHeight := ( GrossSensorOutput - Offsets[ foot ] )
                               * ConversionConstant;

IF ABS( FootHeight ) < 0.1  { inches }
    THEN FootHeight := 0.0;

IF GrossSensorOutput >= 384.0
    THEN FootHeight := 50.0;  { loss-of-signal flag }

END;    { DetermineFootHeight }

{*****************************************************************}

PROCEDURE InitSensorVariables(
        foot                        : Integer;
        Offsets                     : Array6;
    VAR LPFOutput                   : Array6;
    VAR POutput                     : Array6;
    VAR VOutput                     : Array6;
    VAR TrajectoryFlag              : SixCompletions;
        ActualBodyTransform         : Homogeneous;
        ActualFootPositionBody      : SixVectors
                )

CONST   FinalPositionIfOutOfRange = -18.0;

VAR     ActualFootPositionEarth : Vector;
        FootHeight              : Real;

BEGIN

DetermineFootHeight( foot, Offsets, FootHeight );

VOutput[ foot ] := 0.0;

Transform( ActualFootPositionEarth,
           ActualBodyTransform,
           ActualFootPositionBody[ foot ] );

POutput[ foot ] := ActualFootPositionEarth.Z;
LPFOutput[ foot ] := ActualFootPositionEarth.Z - FootHeight;

IF FootHeight = 50.0  { out-of-range flag }
    THEN LPFOutput[ foot ] := FinalPositionIfOutOfRange;

TrajectoryFlag[ foot ] := NotDone;

END;    { InitSensorVariables }
```

```
{*************************************************************************}

PROCEDURE ReInitSensorVariables(
        foot                        : Integer'
        Offsets                     : Array6i
    VAR LPFOutput                   : Array6i
    VAR POutput                     : Array6i
    VAR TrajectoryFlag              : SixCompletions;
        ActualBodyTransform         : Homogeneous;
        ActualFootPositionBody      : SixVectors
                )i

CONST   FinalPositionIfOutOfRange = -18.0i

VAR     ActualFootPositionEarth : Vectori
        FootHeight              : Reali

BEGIN

DetermineFootHeight( foot, Offsets, FootHeight )i
Transform( ActualFootPositionEarth,
           ActualBodyTransform,
           ActualFootPositionBodyC foot ] )i

POutputC foot ] := ActualFootPositionEarth.Zi
LPFOutputC foot ] := ActualFootPositionEarth.Z - FootHeighti

IF FootHeight = 50.0  { out-of-range flag }
    THEN LPFOutputC foot ] := FinalPositionIfOutOfRangei

TrajectoryFlagC foot ] := NotDonei

END;    { ReInitSensorVariables }

{*************************************************************************}

PROCEDURE MoveBody(
        BodyTranslation             : Vectori
        DesiredBodyPosition         : Reali
    VAR DesiredBodyLinearRates      : Vectori
    VAR MoveFlag                    : Completion
                )i

CONST   MaxPositionError = 0.25i{ inches }
        ForwardRate = 3.5i        { inches per second }

VAR     AveragePosition             : Reali
        BodyPositionError           : Reali
        foot                        : Integeri

BEGIN

BodyPositionError := DesiredBodyPosition - BodyTranslation.Xi
```

179

```
{*** EVALUATE BOUNDARY VALUE CONDITION ***}

IF ABS( BodyPositionError ) <= MaxPositionError
    THEN
        MoveFlag := Done

    ELSE
        BEGIN
        MoveFlag := NotDone;

        {*** CALCULATE DESIRED LINEAR BODY RATES ***}

        IF BodyPositionError > 0.0
            THEN
                DesiredBodyLinearRates.X :=  ForwardRate
            ELSE
                DesiredBodyLinearRates.X := -ForwardRate;

        DesiredBodyLinearRates.Y := 0.0;
        DesiredBodyLinearRates.Z := 0.0;

        END; { Else }

END;    { MoveBody }

{*******************************************************************************}

PROCEDURE MaintainFootElevation(
        foot                    : Integer;
        Offsets                 : Array6;
    VAR PlannedFootRate         : Vector
                  );

CONST   Threshold = 2.0;  { inches }

VAR     FootElevation           : Real;

BEGIN

DetermineFootHeight( foot, Offsets, FootElevation );

IF FootElevation < Threshold
  THEN
      PlannedFootRate.Z := 3.0  { inches/second }
  ELSE
      PlannedFootRate.Z := 0.0;

END;    { MaintainFootElevation }

{*******************************************************************************}
```

```
PROCEDURE FinalPositionCalculate(
        dt                         : Real;
        foot                       : Integer;
        ActualBodyTransform        : Homogeneous;
        ActualFootPositionBody     : SixVectors;
        Offsets                    : Array6;
    VAR LPFOutput                  : Array6;
    VAR POutput                    : Array6;
    VAR VOutput                    : Array6;
    VAR ReInitializeFlag           : SixCompletions;
    VAR TrajectoryFlag             : SixCompletions
                );

CONST   Tau = 0.5;   { low-pass filter time constant }
        FinalPositionIfOutOfRange = -18.0;   { inches }

VAR     ActualFootPositionEarth : Vector;
        LPFInput                 : Real;
        DLPFOutput               : Real;
        FootHeight               : Real;

BEGIN  { calculate the filtered value of final earth position }

DetermineFootHeight( foot, Offsets, FootHeight );

IF FootHeight = 50.0
    THEN       { vehicle assumed to be on level ground }
        BEGIN
        LPFOutput[ foot ] := FinalPositionIfOutOfRange;
        ReInitializeFlag[ foot ] := NotDone;
        END

    ELSE IF ( FootHeight <> 50.0 )
        AND ( ReInitializeFlag[ foot ] = NotDone )
          THEN
              BEGIN
              ReInitSensorVariables( foot,
                                     Offsets,
                                     LPFOutput,
                                     POutput,
                                     TrajectoryFlag,
                                     ActualBodyTransform,
                                     ActualFootPositionBody );
              ReInitializeFlag[ foot ] := Done;
              END

    ELSE
        BEGIN
        Transform( ActualFootPositionEarth,
                   ActualBodyTransform,
                   ActualFootPositionBody[ foot ] );

        LPFInput := ActualFootPositionEarth.Z - FootHeight;
        DLPFOutput := (LPFInput - LPFOutput[ foot ]) / Tau;
```

181

```
            LPFOutput[ foot ] := DLPFOutput * dt  +  LPFOutput[ foot ];
            END;                                    { Euler Integration }

END;     { FinalPositionCalculate }

{******************************************************************}

PROCEDURE Trajectory( P0, Pf, V0, Vf, T0, Tf: Real;
                      Vmax, Amax: Real;
                      VAR Acceleration, T1, T2: Real;
                      VAR Possible: Boolean )i

{ A procedure which computes accelerations and timing for a trajectory  }
{ between two points given the position, velocity and time of each      }
{ point.  Velocity and acceleration limits are observed in the          }
{ trajectory generation.                                                }
{                                                                       }
{                  Programmer :  Dennis R. Push                         }
{                                                                       }

VAR     GoodSolution    : Boolean; { Consistant Solution Flag }

        B, C            : Real;    { Quadratic Equation Coeffs. }
        Root            : Real;    { SQRT ( Discriminant ) }
        Discriminant    : Real;    { QUADR. EQ. Discriminant }
        Vs              : Real;    { Velocity at Acceleration Changes }
        Denominator     : Real;    { Intermediate Term }

{***   ***   ***   ***   ***   ***   ***   ***   ***   ***   ***   *** }

    PROCEDURE TimeCheck( Vs, Acceleration: Real; VAR T1, T2: Real;
                         VAR GoodSolution: Boolean )i

{ A procedure which determines if a trajectory can be completed with a  }
{ given acceleration and saturation velocity; it also computes the      }
{ times at which the acceleration state is changed.                     }

    BEGIN { TimeCheck }

    IF ABS( Acceleration ) < 0.001
        THEN
            BEGIN
            IF ( ABS( Pf-P0 - V0*(Tf-T0) ) < 0.01 ) { Inches }
                AND ( ABS( Vf - V0 ) < 0.1 )
                THEN
                    BEGIN
                    GoodSolution := True;
                    T1 := T0;
                    T2 := Tf;
                    END
                ELSE
                    GoodSolution := False;
                { END IF }
            END { THEN }
```

182

```
                ELSE
                    BEGIN
                    T1 := ( Vs - V0 ) / Acceleration + T0;
                    T2 := Tf - ( Vs - Vf ) / Acceleration;
                    IF        ( T1 >= T0 - 0.01 )
                        AND ( T2 >= T1 - 0.01 )
                        AND ( Tf >= T2 - 0.01 )
                        THEN GoodSolution := True
                        ELSE GoodSolution := False
                    END;

            < END IF Acceleration >

        END; < TimeCheck >

<***   ***   ***   ***   ***   ***   ***   ***   ***   ***   ***   *** >

BEGIN  < Trajectory >

B := -2.0 * ( Pf - P0 ) / ( Tf - T0 );
C := ( Pf - P0 ) * ( Vf + V0 ) / ( Tf - T0 ) - ( Vf*Vf + V0*V0 ) / 2.0;

Discriminant := B*B - 4.0*C;

IF Discriminant = 0.0
    THEN Root := 0.0
    ELSE Root := SQRT( Discriminant );

Vs := ( -B + Root ) / 2.0;
Acceleration := ( 2.0 * Vs - Vf - V0 ) / ( Tf - T0 );
TimeCheck( Vs, Acceleration, T1, T2, GoodSolution );

IF GoodSolution = False
    THEN
        BEGIN
        Vs := ( -B - Root ) / 2.0;
        Acceleration := ( 2.0 * Vs - Vf - V0 ) / ( Tf - T0 );
        TimeCheck( Vs, Acceleration, T1, T2, GoodSolution );
        END; < THEN >

    < END IF GoodSolution >

IF ( ABS( Acceleration ) < Amax ) AND ( ABS( Vs ) > Vmax )
    THEN < Try Trajectory Type 2 >
        BEGIN

        IF Vs > 0.0
            THEN Vs := Vmax
            ELSE Vs := -Vmax;
```

183

```
            Denominator := ( Tf - T0 ) * Vs - Pf + P0;   { Inches }
            IF ABS( Denominator ) <= 0.01
                THEN
                    Acceleration := 0.0
                ELSE
                    Acceleration := (SQR(Vs-V0)+SQR(Vs-Vf))/Denominator/2.0;

            TimeCheck( Vs, Acceleration, T1, T2, GoodSolution );

            END; { THEN }

        { END IF }

    IF ( GoodSolution = True ) AND ( ABS( Acceleration ) <= Amax + 0.1 )
        THEN
            Possible := True
        ELSE
            BEGIN
            Possible := False;
            IF Acceleration > 0.0
                THEN
                    Acceleration := Amax
                ELSE
                    Acceleration := -Amax;
            END;

END;    { Trajectory }

{*************************************************************************}

PROCEDURE FindPlaceDownRate(
        dt                      : Real;
        foot                    : Integer;
        ActualBodyTransform     : Homogeneous;
        FootContact             : Boolean;
    VAR Downrate                : Real;
    VAR LPFOutput               : Array6;
    VAR POutput                 : Array6;
    VAR VOutput                 : Array6;
    VAR TrajectoryFlag          : SixCompletions;
    VAR FinalTime               : Array6;
    VAR RealTime                : Real
                );

CONST   Vmax = 5.0;     { inches per second }
        Amax = 10.0;    { inches per second per second }
```

184

```
VAR       Deltap                    : Real;
          Tmin                      : Real;
          TC                        : Real;
          Tf                        : Real;
          PO                        : Real;
          Pf                        : Real;
          Vs                        : Real;
          VO                        : Real;
          Vf                        : Real;
          T1                        : Real;
          T2                        : Real;
          Acc                       : Real;

          Poss                      : Boolean;

          DesiredFootRateEarth      : SixVectors;
          DesiredFootRateBody       : SixVectors;
          InverseBodyRotation       : Matrix;

BEGIN

PO := POutput[ foot ];
Pf := LPFOutput[ foot ];

VO := VOutput[ foot ];
Vf := 0.0;   { inch/second } { desired final velocity of all less }

TO := RealTime;
Tf := FinalTime[ foot ];

Deltap := Pf - PO;

IF ( TrajectoryFlag[ foot ] = Done )
  OR (( ABS( Deltap ) < 4.0 ) AND ( ABS( VOutput[ foot ] ) < 2.05 ))
        THEN            { begin constant velocity search phase }
            BEGIN
            TrajectoryFlag[ foot ] := Done;
            DownRate := -2.0;    { inches/second }
            VOutput[ foot ] := DownRate;
            IF FootContact = True   THEN   VOutput[ foot ] := 0.0;
            POutput[ foot ] := POutput[ foot ] + VOutput[ foot ]* dt;
            END

        ELSE
            BEGIN   {  GENERATE POSITION AND VELOCITY TRAJECTORIES  }

            Trajectory(PO,Pf,VO,Vf,TO,Tf,Vmax,Amax,Acc,T1,T2,Poss);

            IF TO >= T2
              THEN Acc := -Acc    { decelerate }
              ELSE IF TO >= T1
                    THEN Acc := 0.0;   { move @ maximum velocity }

            IF FootContact = True   THEN   Acc := 0.0;
```

185

```
          VOutput[ foot ] := VO  +  dt * Acc;  { Euler integration}

          POutput[ foot ] := PO +  dt/2.0 * (VO + VOutput[ foot ]);
                                      { trapezoidal integration }

          DesiredFootRateEarth[ foot ].X := 0.0;
          DesiredFootRateEarth[ foot ].Y := 0.0;
          DesiredFootRateEarth[ foot ].Z := VOutput[ foot ];

          { Convert Desired Foot Velocity to Body Coordinates }

          Transpose( InverseBodyRotation,
                      ActualBodyTransform.Rotation);

          MatrixMult( DesiredFootRateBody[ foot ],
                      InverseBodyRotation,
                      DesiredFootRateEarth[ foot ]);

          DownRate := DesiredFootRateBody[ fout ].Z;

          END;  { IF ( ) AND ( ) }

   END;    { FindPlaceDownRate }

{**************************************************************************}

PROCEDURE MoveFootForKeith(
          Offsets                  : Array6;
          foot                     : Integer;
          FootContact              : Boolean;
          DesiredFootPosition      : Vector;
          dt                       : Real;
          DelT                     : Real;
      VAR ActualBodyTransform      : Homogeneous;
      VAR ActualFootPositionBody   : SixVectors;
      VAR ZeroForce                : Vector;
      VAR PlannedFootRate          : Vector;
      VAR FootRateType             : RateType;
      VAR MoveFootFlag             : Completion;
      VAR LPFOutput                : Array6;
      VAR POutput                  : Array6;
      VAR VOutput                  : Array6;
      VAR RealTime                 : Real;
      VAR ReInitializeFlag         : SixCompletions;
      VAR TrajectoryFlag           : SixCompletions;
      VAR FinalTime                : Array6;
      VAR InitialTime              : Array6
                     );

CONST   LiftRate = 3.5;         { inches per second }
        ReturnRate = 3.5;       { inches per second }
        MaxError = 0.1;         { inches }
```

186

```
        PlacePoint = SixVectors(( 28.75,  24.00, -10.00 ),
                                ( 28.75, -24.00, -10.00 ),
                                (  6.00,  24.00, -10.00 ),
                                (  6.00, -24.00, -10.00 ),
                                (-16.75,  24.00, -10.00 ),
                                (-16.75, -24.00, -10.00 ));

VAR*    DeltaPosition    : Vector;    ( Intermediate position vector )
        Distance         : Real;      ( Distance to PlacePoint )
        MotionDirection  : Vector;    ( Unit vector toward PlacePoint )
        TemporaryForce   : SixVectors;
        DownRate         : Real;
        FootElevation    : Real;
        FootHeight       : Real;

BEGIN  ( MoveFootForKeith )

(*** EVALUATE BOUNDARY VALUE CONDITIONS ***)

CASE MoveFootState[ foot ] OF

    Ground:    BEGIN
               MoveFootState[ foot ] := LiftOff;
               MoveFootFlag := NotDone;
               END;

    LiftOff:   BEGIN
               IF DesiredFootPosition.Z >= PlacePoint[ foot ].Z
                   THEN  ( foot is at top of trajectory )
                       BEGIN
                       MoveFootState[ foot ] := Return;

                       ForceFeedback( TemporaryForce );

                       ZeroForce := TemporaryForce[ foot ];

                       VectSub( DeltaPosition,
                                PlacePoint[foot], DesiredFootPosition );

                       Distance := Magnitude( DeltaPosition );
                       END; ( if desired foot position )
               END;

    Return:    BEGIN
               VectSub( DeltaPosition,
                        PlacePoint[foot], DesiredFootPosition );

               Distance := XYMagnitude( DeltaPosition );
```

187

```
                    IF Distance <= MaxError
                        THEN  { foot ready for placing }
                            BEGIN
                            MoveFootState[ foot ] := PlaceDown;
                            InitSensorVariables( foot,
                                                 Offsets,
                                                 LPFOutput,
                                                 POutput,
                                                 VOutput,
                                                 TrajectoryFlag,
                                                 ActualBodyTransform,
                                                 ActualFootPositionBody );

                            InitialTime[ foot ] := RealTime;
                            FinalTime[ foot ] := RealTime + DelT;
                            END;

                    END;

    PlaceDown:  IF FootContact
                    THEN
                        BEGIN
                        MoveFootState[ foot ] := Ground;
                        MoveFootFlag := Done;
                        END;

    END; { CASE MoveFootState }

{*** COMPUTE FOOT VELOCITY COMMAND ***}

CASE MoveFootState[ foot ] OF

    Ground:     BEGIN
                PlannedFootRate := Zero;
                FootRateType := Relative;
                END;

    LiftOff:    BEGIN
                PlannedFootRate.X := 0.0;
                PlannedFootRate.Y := 0.0;
                PlannedFootRate.Z := LiftRate;

                FootRateType := Relative;
                END; { LiftOff }

    Return:     BEGIN
                VectDiv( MotionDirection,
                              DeltaPosition, Distance );

                VectMult( PlannedFootRate,
                              ReturnRate, MotionDirection );
```

188

```
              MaintainFootElevation( foot,
                                      Offsets,
                                      PlannedFootRate );

              FootRateType := Absolute;

              END; ( Return )

    PlaceDown:  BEGIN

              FinalPositionCalculate( dt,
                                      foot,
                                      ActualBodyTransform,
                                      ActualFootPositionBody,
                                      Offsets,
                                      LPFOutput,
                                      POutput,
                                      VOutput,
                                      ReInitializeFlag,
                                      TrajectoryFlag );

              FindPlaceDownRate( dt,
                                 foot,
                                 ActualBodyTransform,
                                 FootContact,
                                 Downrate,
                                 LPFOutput,
                                 POutput,
                                 VOutput,
                                 TrajectoryFlag,
                                 FinalTime,
                                 RealTime );

              PlannedFootRate.X := 0.0;
              PlannedFootRate.Y := 0.0;
              PlannedFootRate.Z := DownRate;

              FootRateType := Relative;
              END; ( PlaceDown )

     END; ( CASE MoveFootState )

END;    ( MoveFootForKeith )

(*****************************************************************************)

PROCEDURE SoftStep;

VAR     foot                    : Integer;
        move                    : Integer;

BEGIN   ( Use Sensor Data During Hexapod Locomotion )
```

189

```
{*** CALL THE APPROPRIATE MOTION SUBROUTINE ***}

CASE SoftStepState OF

    ReadyToGo:

        BEGIN
        DesiredBodyLinearRates := Zero;
        DesiredBodyAngularRates := Zero;

        FOR foot := 1 to 6 DO
            BEGIN
            PlannedFootRate[ foot ] := Zero;
            FootRateType[ foot ] := Relative;
            END;

        END;

    Step1:

        BEGIN
        DesiredBodyLinearRates := Zero;
        DesiredBodyAngularRates := Zero;

        FOR foot := 1 to 6 DO
            IF ( foot IN [ 1, 4, 5 ] )
              AND ( MoveFootFlag[ foot ] = NotDone )
                  THEN
                      MoveFootForKeith( Offsets,
                                        foot,
                                        FootContact[ foot ],
                                        DesiredFootPosition[ foot ],
                                        dt,
                                        DelT,
                                        ActualBodyTransform,
                                        ActualFootPositionBody,
                                        ZeroForce[ foot ],
                                        PlannedFootRate[ foot ],
                                        FootRateType[ foot ],
                                        MoveFootFlag[ foot ],
                                        LPFOutput,
                                        POutput,
                                        VOutput,
                                        RealTime,
                                        ReInitializeFlag,
                                        TrajectoryFlag,
                                        FinalTime,
                                        InitialTime )

                  ELSE
                      BEGIN
                      PlannedFootRate[ foot ] := Zero;
                      FootRateType[ foot ] := Relative;
                      END;
        END;
```

190

```
Move1st:

    BEGIN
    MoveBody(                        DesiredBodyTransform.Translation,
                                     8.0,
                                     DesiredBodyLinearRates,
                                     MoveBodyFlag[1] );

    DesiredBodyAngularRates := Zero;

    FOR foot := 1 to 6 DO
        BEGIN
        PlannedFootRate[ foot ] := Zero;
        FootRateType[ foot ] := Relative;
        END;

    END;

Step2:

    BEGIN
    DesiredBodyLinearRates := Zero;
    DesiredBodyAngularRates := Zero;

    FOR foot := 1 TO 6 DO
        IF ( foot IN [ 2, 3, 6 ] )
          AND ( MoveFootFlag[ foot ] = NotDone )
              THEN
                  MoveFootForKeith( Offsets,
                                    foot,
                                    FootContact[ foot ],
                                    DesiredFootPosition[ foot ],
                                    dt,
                                    DelT,
                                    ActualBodyTransform,
                                    ActualFootPositionBody,
                                    ZeroForce[ foot ],
                                    PlannedFootRate[ foot ],
                                    FootRateType[ foot ],
                                    MoveFootFlag[ foot ],
                                    LPFOutput,
                                    POutput,
                                    VOutput,
                                    RealTime,
                                    ReInitializeFlag,
                                    TrajectoryFlag,
                                    FinalTime,
                                    InitialTime )

              ELSE
                  BEGIN
                  PlannedFootRate[ foot ] := Zero;
                  FootRateType[ foot ] := Relative;
                  END;
    END;
```

191

```
Move2nd:

    BEGIN
    MoveBody(                       DesiredBodyTransform.Translation,
                                    16.0,
                                    DesiredBodyLinearRates,
                                    MoveBodyFlag[2] );

    DesiredBodyAngularRates := Zero;

    FOR foot := 1 to 6 DO
        BEGIN
        PlannedFootRate[ foot ] := Zero;
        FootRateType[ foot ] := Relative;
        END;

    END;

Step3:

    BEGIN
    DesiredBodyLinearRates := Zero;
    DesiredBodyAngularRates := Zero;

    FOR foot := 1 to 6 DO
        IF ( foot IN [ 1, 4, 5 ] )
          AND ( MoveFootFlag[ foot ] = NotDone )
                THEN
                    MoveFootForKeith( Offsets,
                                      foot,
                                      FootContact[ foot ],
                                      DesiredFootPosition[ foot ],
                                      dt,
                                      DelT,
                                      ActualBodyTransform,
                                      ActualFootPositionBody,
                                      ZeroForce[ foot ],
                                      PlannedFootRate[ foot ],
                                      FootRateType[ foot ],
                                      MoveFootFlag[ foot ],
                                      LPFOutput,
                                      POutput,
                                      VOutput,
                                      RealTime,
                                      ReInitializeFlag,
                                      TrajectoryFlag,
                                      FinalTime,
                                      InitialTime )

                ELSE
                    BEGIN
                    PlannedFootRate[ foot ] := Zero;
                    FootRateType[ foot ] := Relative;
                    END;
    END;
```

192

```
Move3rd:

    BEGIN
    MoveBody(                    DesiredBodyTransform.Translation,
                                 24.0,
                                 DesiredBodyLinearRates,
                                 MoveBodyFlag[3] );

    DesiredBodyAngularRates := Zero;

    FOR foot := 1 to 6 DO
        BEGIN
        PlannedFootRate[ foot ] := Zero;
        FootRateType[ foot ] := Relative;
        END;

    END;

Step4:

    BEGIN
    DesiredBodyLinearRates := Zero;
    DesiredBodyAngularRates := Zero;

    FOR foot := 1 to 6 DO
        IF ( foot IN [ 2, 3, 6 ] )
          AND ( MoveFootFlag[ foot ] = NotDone )
            THEN
                MoveFootForKeith( Offsets,
                                  foot,
                                  FootContact[ foot ],
                                  DesiredFootPosition[ foot ],
                                  dt,
                                  DelT,
                                  ActualBodyTransform,
                                  ActualFootPositionBody,
                                  ZeroForce[ foot ],
                                  PlannedFootRate[ foot ],
                                  FootRateType[ foot ],
                                  MoveFootFlag[ foot ],
                                  LPFOutput,
                                  POutput,
                                  VOutput,
                                  RealTime,
                                  ReInitializeFlag,
                                  TrajectoryFlag,
                                  FinalTime,
                                  InitialTime )

            ELSE
                BEGIN
                PlannedFootRate[ foot ] := Zero;
                FootRateType[ foot ] := Relative;
                END;
```

193

```
                END;

            END; ( CASE SoftStepState )

    (*** COMPUTE NEW BODY TRANSFORM ***)

    TransIntegrate( DesiredBodyTransform,
                    DesiredBodyLinearRates,
                    DesiredBodyAngularRates,
                    dt );

    (*** DETERMINE NEW STATE BASED ON MOTION BOUNDARY VALUES ***)

    CASE SoftStepState OF

        ReadyToGo:

            BEGIN
            SoftStepState := Step1;
            SoftStepFlag := NotDone;
            FOR foot := 1 to 6 DO
                MoveFootFlag[ foot ] := NotDone;
            FOR move := 1 to 3 DO
                MoveBodyFlag[ move ] := NotDone;
            END;

        Step1:

            BEGIN
            Step1Flag := Done;
            FOR foot := 1 to 6 DO
                IF ( foot IN [ 1, 4, 5 ] )
                   AND ( MoveFootFlag[ foot ] = NotDone )
                     THEN
                         Step1Flag := NotDone;
            IF Step1Flag = Done
                 THEN
                     SoftStepState := Move1st;

            END; ( Step1 )

        Move1st:

            BEGIN
            IF MoveBodyFlag[1] = Done
                 THEN
                     BEGIN
                     SoftStepState := Step2;
                     For foot := 1 to 6 DO
                         MoveFootFlag[ foot ] := NotDone;
                     END;

            END; ( Move1st )
```

194

```
Step2:

    BEGIN
    Step2Flag := Done;
    FOR foot := 1 to 6 DO
        IF ( foot IN [ 2, 3, 6 ] )
            AND ( MoveFootFlag[ foot ] = NotDone )
                THEN
                    Step2Flag := NotDone;

    IF Step2Flag = Done
        THEN
            SoftStepState := Move2nd;

    END; ( Step2 )

Move2nd:

    BEGIN
    IF MoveBodyFlag[2] = Done
        THEN
            BEGIN
            SoftStepState := Step3;
            For foot := 1 to 6 DO
                MoveFootFlag[ foot ] := NotDone;
            END;

    END; ( Move2nd )

Step3:

    BEGIN
    Step3Flag := Done;
    FOR foot := 1 to 6 DO
        IF ( foot IN [ 1, 4, 5 ] )
            AND ( MoveFootFlag[ foot ] = NotDone )
                THEN
                    Step3Flag := NotDone;

    IF Step3Flag = Done
        THEN
            SoftStepState := Move3rd;

    END; ( Step3 )

Move3rd:

    BEGIN
    IF MoveBodyFlag[3] = Done
        THEN
            BEGIN
            SoftStepState := Step4;
            For foot := 1 to 6 DO
                MoveFootFlag[ foot ] := NotDone;
            END;
```

195

```
        END; { Move3rd }

    Step4:

        BEGIN
        Step4Flag := Done;
        FOR foot := 1 to 6 DO
            IF ( foot IN [ 2, 3, 6 ] )
                AND ( MoveFootFlag[ foot ] = NotDone )
                THEN
                    Step4Flag := NotDone;

        IF Step4Flag = Done
            THEN
                BEGIN
                SoftStepState := ReadyToGo;
                SoftStepFlag := Done;
                END;

        END; { Step4 }

    END; { Case SoftStepState }

END;    { SoftStep }

{******************************************************************************}
```

196

REFERENCES

1. Kay, L., "An Ultrasonic Sensing Probe as a Mobility Aid For the Blind", Ultrasonics, Vol. 2, April-June, 1964.

2. Anon, "A Novel Prosthetic Sonar", Ultrasonics, Vol. 15, No. 3, May, 1977.

3. Boys, J.T., Strelow, E.R., Clark, G.R.S., "A Prosthetic Aid For a Developing Blind Child", Ultrasonics, Vol. 17, No. 1, January, 1979.

4. Orin, D.E., "Supervisory Control of a Multilegged Robot", The International Journal of Robotics Research, Vol. 1, No. 1, Spring, 1982.

5. Klein, C.A., Olson, K.W., Pugh, D.R., "Use of Force and Attitude Sensors for Locomotion of a Legged Vehicle Over Irregular Terrain", The International Journal of Robotics Research, Vol. 2, No. 2, Summer, 1983.

6. Pugh, D.R., An Autopilot For a Terrain-Adaptive Hexapod Vehicle, M.S. thesis, Ohio State University, Columbus, OH, September, 1982.

7. Briggs, R.L., Real-Time Digital Control of an Electrically Powered Vehicle Leg Using Vector Force Feedback, M.S. thesis, Ohio State University, Columbus, OH, December, 1977.

8. Buckett, J.R., Design of an On-Board Electronic Joint Control System for a Hexapod Vehicle, M.S. thesis, Ohio State University, Columbus, OH, March, 1977.

9. Umetani, Y., Hirose, S., "Biomechanical Study of Active Cord Mechanism with Tactile Sensors", presented at 6th International Symposium on Industrial Robots, England, 1976.

10. Rasmussen, J., "Proximity Switches", presented at the 3rd International Conference on Automated Inspection and Product Control, England, April, 1978.

11. Fosler, K., Parker, G.A., Fluidics-Components and Circuits, Wiley Interscience, London, England, 1970.

197

12. Hanafusa, H., Asada, H., "An Adaptive Control of Robot Hand Manipulators Equipped With Pneumatic Proximity Sensors", presented at 6th International Symposium on Industrial Robots, England, 1976.

13. Bulthuis, K., Carasso, M., Heemskerk, J., Kivits, P., "Ten Billion Bits on a Disk", IEEE Spectrum, Vol. 16, No. 8, August, 1979.

14. Anon., LM1812 Ultrasonic Transceiver notes, Linear Databook, National Semiconductor Corporation, Santa Clara, CA, 1982.

15. Biber, C., Ellin, S., Shenk, E., Stempeck, J., "The Polaroid Ultrasonic Ranging System", presented at the 67th Convention of the Audio Engineering Society, New York, October 31, 1980. .

16. Bauzil, G., Briot, M., Ribes, P., "A Navigation Sub-System Using Ultrasonic Sensors for the Mobile Robot Hilare", presented at the Symposium on Computer Vision and Sensory Controls, Stratford-Upon-Avon, England, 1981.

17. Morander, K.E., "The Optocator, a High Precision, Non-Contacting System for Dimension and Surface Measurement and Control", presented at the 5th International Conference on Automated Inspection & Product Control, Stuttgart, Germany, June, 1980.

18. Tsai, S.J., An Experimental Study of a Binocular Vision System for Rough Terrain Locomotion of a Hexapod Walking Robot, PhD dissertation, Ohio State University, Columbus, OH, May, 1983.

19. Espiau, B., Catros, J.Y., "Use of Optical Reflectance Sensors in Robotics Applications", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-10, No. 12, December, 1980.

20. Zuk, D.M., Dell'eva M.L., Three-Dimensional Vision System for the Adaptive Suspension Vehicle, Final Report, Environmental Research Institute of Michigan, Ann Arbor, MI, January, 1983.

21. Bishel, R.A., The Design and Development of an Ultra-Short Range Radar for Use in a Lateral Control System for Automated Ground Transportation, M.S. thesis, Ohio State University, Columbus, OH, 1980.

22. Frederiksen, T.M., Howard, W.M., "A Single-Chip Monolithic Sonar System", IEEE Journal of Solid-State Circuits, Vol. SC-9, No. 6, December, 1974.

23. Anon., LM1812 Ultrasonic Transceiver notes, Linear Databook, National Semiconductor Corporation, Santa Clara, CA, 1981.

198

24. Canali, C., De Cicco, G., et al, "A Temperature Compensated Ultrasonic Sensor Operating in Air for Distance and Proximity Measurements", IEEE Transactions on Industrial Electronics, Vol. IE-29, No. 4, November, 1982.

25. Anon., DR11-C General Device Interface Manual, Digital Equipment Corporation, Maynard, Massachusetts, 1974.

2€. Maney, J.J., A Comparative Study of Real-Time Computer Control Techniques for a Multi-Jointed Linkage Mechanism, M.S. ᴟesis, Ohio State University, Columbus, OH, June, 1980.

27. Pugh, D.R., Partitioning of Body Motion and Force Distribution Control Spaces in the Control of a Statically Stable Multilegged Vehicle, Digital Systems Laboratory Technical Note #27, Department of Electrical Engineering, Ohio State University, Columbus, OH, August, 1983.

28. Barrientos, C.E., Development of a Multiple Microprocessor-Based Controller For a Legged Locomotion Process Using Interactive Design Tools, M.S. thesis, The Ohio State University, Columbus, OH, December, 1982.